UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

AUTOMATIC CALCULATION OF: COST, DISTANCE AND DURATION OF INTERNATIONAL ROAD FREIGHT TRANSPORTATION

Wilson Miguel Ferreira Pontes Cardoso

Mestrado em Matemática Aplicada à Economia e Gestão

Projecto orientado por:
Prof. PhD João Miguel Paixão Telhada

2017

Acknowledgments

Prof. PhD João Paixão Telhada; For being my advisor, always pushing me to improve my work. Being someone with whom I wanted to work since the first class I had with. Stimulating my mind with every meeting we had.

Prof. PhD Maria Teresa Alpuim; For accepting me on this course, always ready to receive me, helping me along the way, guiding me with any issue I had.

Prof. PhD Inês Marques Proença; For seeing in me the aptitude and for suggesting me to enroll on this challenge.

Prof. PhD André Vieira Fonseca; For helping me remember how must I enjoy mathematics, that was the beginning of this long ride.

MSc Alexandre Braamcamp Pereira; A friend who was always a chat away from helping me with my code when creativity was a miss.

And above all, my Li. Without whom I would not have been able to endure and finish this challenge. This was my desert and my mountain, and you helped me cross it and climb it. You did not let me give up, and you always encouraged me when I doubted myself. I could not have done it without your support and understanding.

I thank You, with all that I am.

Abstract

The purpose of this work is to develop a program that will produce several different schedules for freight transport drivers, respecting several constraints, for multiple destinations. Once this information is produced, it is then selected the best global solution that satisfies the demand of the different destinations with minimum costs.

Firstly it will be presented a contextualization of the freight transportation reality in Europe: analyzing the economic impact on several countries GDP; the importance for labour; the hourly costs, and laws associated.

Then there will be explained the different constraints. Constraints imposed by the European Union and their reasons. Locations' related constraints. And constraints regarding the company's country of origin.

There will also be presented the required inputs, in other words, the information necessary for the program to work.

Next, it will be given a description of the approach taken to build the program. How the costs were taken into consideration. How the heuristic was built to produce each driver's different schedules. Then there are mentioned the advantages in typifying drivers. And the use of multiple drivers for the same delivery.

Finally it is presented an example, where a company with head quarters in Portugal has to satisfy the demand for 3 destinations throughout Europe. Using single drivers, and/or multiple drivers per delivery with an equal weight for the direct costs (hourly rates) and indirect costs (delay costs). This sensitivity analysis (alpha) will also be evaluated.

Keywords: Logistics, Driver, Schedule, Destination, Constraints, Heuristic, Optimization.

Resumo

O objectivo deste trabalho é desenvolver um programa que produza vários calendários diferentes para condutores de transporte de mercadorias, respeitando as várias restrições, para múltiplos destinos. Assim que esta informação seja produzida, será seleccionada a melhor opção global que satisfaça a procura dos diferentes destinos com o mínimo custo.

Inicialmente será apresentada uma contextualização da realidade do transporte de mercadorias na Europa: analisando o impacto económico no PIB de vários países; a importância no mercado laboral; O custo hora, e as leis associadas a este.

Então serão explicadas a diferentes restrições: restrições impostas pela União Europeia e as suas razões: restrições relacionadas com a localização; e restrições a respeito do país de origem da empresa.

Serão também apresentados os inputs necessários, por outras palavras, a informação necessaria para que o programa funcione.

De seguida será dada uma descrição da abordagem seguida na construção do programa: como os custos foram considerados; como a heurística foi construida para produzir os diferentes calendários de cada condutor; são mencionadas as vantagens em tipificar condutores; e o uso de múltiplos condutores para a mesma entrega.

Finalmente é apresentado um exemplo, onde uma empresa com sede em Portugal tem que satisfazer a procura de 3 destinos da Europa, recorrendo a condutores individuais, e/ou pares de múltiplos condutores por entrega. Com um peso igual entre custos directos (custos hora) e custos indirectos (custos por demora). Esta análise de sensibilidade (alpha) será também avaliada.

Dado que os softwares utilizados estavam em Inglês, foi usado o ”.” como casa decimal.

Palavras-chave: Logística, Condutor, Calendário, Restrições, Heurística, Optimização.

Contents

List of Figures

List of Tables

1: Introduction

Transporting goods internationally by truck is to this day, still the cheapest, most efficient and robust way of delivery. However this does not imply perfection. Not only, road transportation is not always the best solution, as when it is, several different solutions are available, choosing the best one is then a challenge.

1.1 Problem

Finding what is the best, (or the least bad) solution, is a problem. What criteria to focus on? Minimizing variable costs with drivers? Minimizing trip duration, destination arriving moment, return moment, number of trips, idle hours for truck tractors? Or perhaps attempting to get a mix solution involving multiple criteria. But then what criteria should weight more?

These are just a few of the many existing problems when trying to reach a good solution. The more criteria, the greedy the solution's goal, the harder it is to calculate and reach it in useful time. An imperfect solution will always trump a perfect solution that is reached too late.

1.2 Objectives

The global goal is to develop an automatic mechanism that provides answers to the manager and decision maker, regarding the allocation of drivers in road freight transportation. So as to facilitate the process of determining the costs of services, in function of a varied set of data, such as the day of departure and/or the number of drivers to be used. Some examples and a simulation will be presented, and analyzed.

1.3 Document structure

The following, chapter 2, will be a brief representation of the road freight transportation's context in Europe. Chapter 3, will focus on describing the problem in the allocation of drivers, and on the driving rules imposed by the European Union and the country where a fictional company, used as an example, has its base of operations.

The 4$^{\text{th}}$ chapter will clarify on the mechanism's design and some examples will be provided and analyzed. In the 5$^{\text{th}}$ chapter there will be analyzed; a global overview of the mechanism and its full potential, a large example, and the mechanism's drawbacks.

Finally in the 6$^{\text{th}}$ chapter it will be argued what should have been achieved, what could have been approached differently, and what would be interesting to explore in the future.

2: Context

Road freight transportation has been possible since humanity has discovered the wheel. It has an important role in society, transporting every type of goods, be that, medicine, fuel or food, from raw resources like stone, minerals or coal, to technology like computers, smart phones or airplane parts. And also human beings, pets and live stock. Anything and everything is transportable by road.

By the beginning of the XX century freight transportation was made mostly by rail. It was with the development of road systems, like highways, and the heavy industrialization of the automobile that road transportation began to grow, being now the most used form of transportation.

The impact in the economy in undeniable. In many European countries road freight transportation has a share of 2% GDP, with countries going up to 3 times that percentage. Most goods are transported by road, even if most of the trip isn't done by road, hardly any good transported can avoid being transported this way. Even if only for the final part of the trip, an item transported by sea, air or rail, will not reach its final destination this way.

Freight transportation employs millions of workers, such as; drivers, vehicle manufactures, maintenance, and regulation of the sector. Some countries have specific rules that are applied within their borders, these have a direct impact on these countries' companies, and on the foreign companies and professionals operating there. But in the case of Europe this also creates a degree of inequality among EU members.

In this chapter there will be presented a short view of the economic impact of road freight transportation and the impact on labour regarding specific rules on some countries.

To measure the amount of goods transported, it is used the tonne per kilometer (tkm) ratio. This represents the transport of 1 tonne of goods (gross weight) over a distance of 1 kilometer [16]. A 20 tkm, means that 20 tons were driven for 1 kilometer, or that 1 ton was transported for 20 kilometers. If 5 tons were transported for 11 kilometers, then that would mean that a total of 55 tkm were transported.

2.1 Economics of freight transportation

Freight transportation is an important component of the EU economy and society, having goods being produced or manufactured in a country and then being transported to the whole continent makes the union more universal. Economically, it is far more efficient to have specialized areas for producing goods and then ship these goods throughout everyone else, than to have every country produce its own product in a less efficient way.

In 2015 a 163,806 Million kilometers were driven throughout the 28 countries of Euro group (EU-28). According to the statistical pocketbook of 2016, published by the European Commission [15], it is estimated that in 2014 the transport services shared amounts of 5.1% of total gross value added (that is of €633 billion at current prices) in the EU-28. For the same year, 5.1% of the total workforce in the 28 countries of the EU worked in the sector of transport and storage, which translates into around 11 million persons.

The total consumption of private households (in the EU-28) on transport-related items in 2014, is estimated to be of €1,001 billion (roughly a 13% of private households total consumption) [15]. More than 52% of this expenditure was on private transport, such as vehicle fuel. €265 billion were spent on the purchase of private vehicles while the remaining 21% were used on public transports, like bus, train, and plane tickets, etc..

In Figure 2.1 it is presented the growth of GDP in Europe (green line) indexed to the year of 1995, and also the growth in goods transported in Europe (red line) in tonnes per kilometer. Both growths are very similar, after the crisis in 2008 both suffered decays with goods transported having a most pronounced decay. The amount of goods transported throughout Europe is dependent on the economy, the more stable the economy the more goods are being shipped, with less purchasing power, less goods are transported, especially superfluous goods. With more goods being transported due to more demand, the sector restarts to growth which in turn contributes to the economic growth. After 2012 both growths seem to resume their original trend.

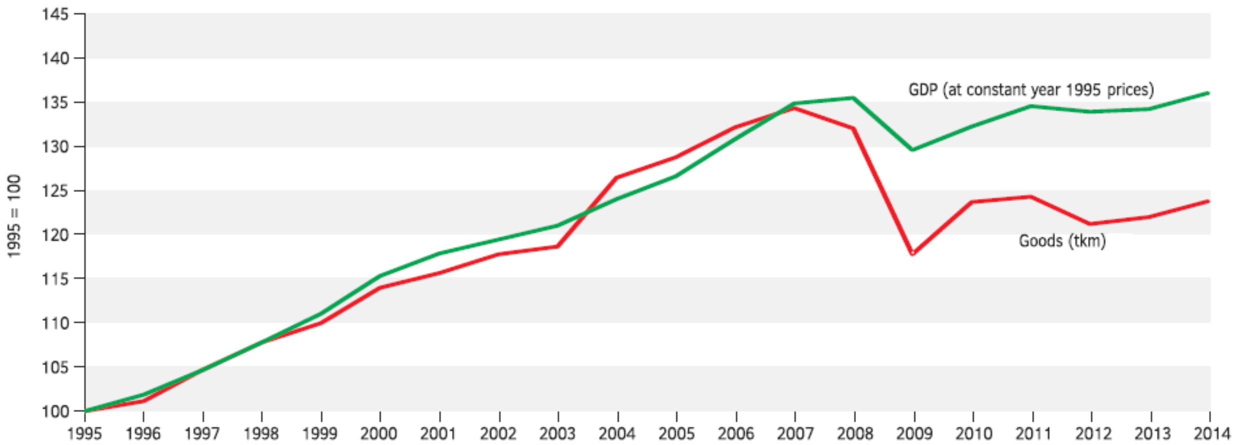

Figure 2.1: Impact of the road freight transport in EU in the years 1995-2014.

Source: The Impact of Regulation of the Road Transport Sector on Entrepreneurship and Economic Growth in the European Union, Motor Transport Institute [13].

Figure 2.2 shows the amount of tons transported after the economic crisis. There has been a decrease since 2008, but the later years seem to start to recover from such crisis. In the year of 2015, of the 61% of the EU market, 3,459 million tons were from Germany alone, which are more than the 2nd and 3rd countries combined. Curiously, road freight transportation does not have such a big share of GDP in Germany as it has in the rest of the EU-28 [13], with exception of countries that are islands (UK, Ireland, Cyprus and Malta), that are unable to rely on road transportation.

By 2050, it can be expected that the demand for transport services will double in Europe [13]. Figure 2.3 shows that road transportation (blue line) is not only the most used option but the most growing method

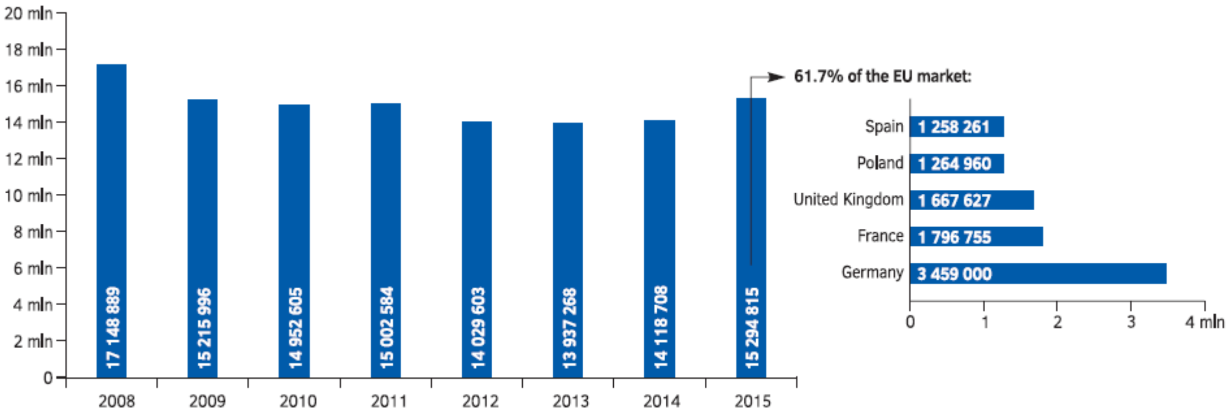

Figure 2.2: Road freight transport in EU in the years 2008 to 2015 (in thousand tons).

Source: The Impact of Regulation of the Road Transport Sector on Entrepreneurship and Economic Growth in the European Union, Motor Transport Institute [13].

of transportation, with a trend indicating that this growth will continue in the near future (black line). A total of 3524 billion tkm goods were estimated to be transported in 2014 within the EU-28 [15]. Out of this amount, 49% was from road transportation alone.

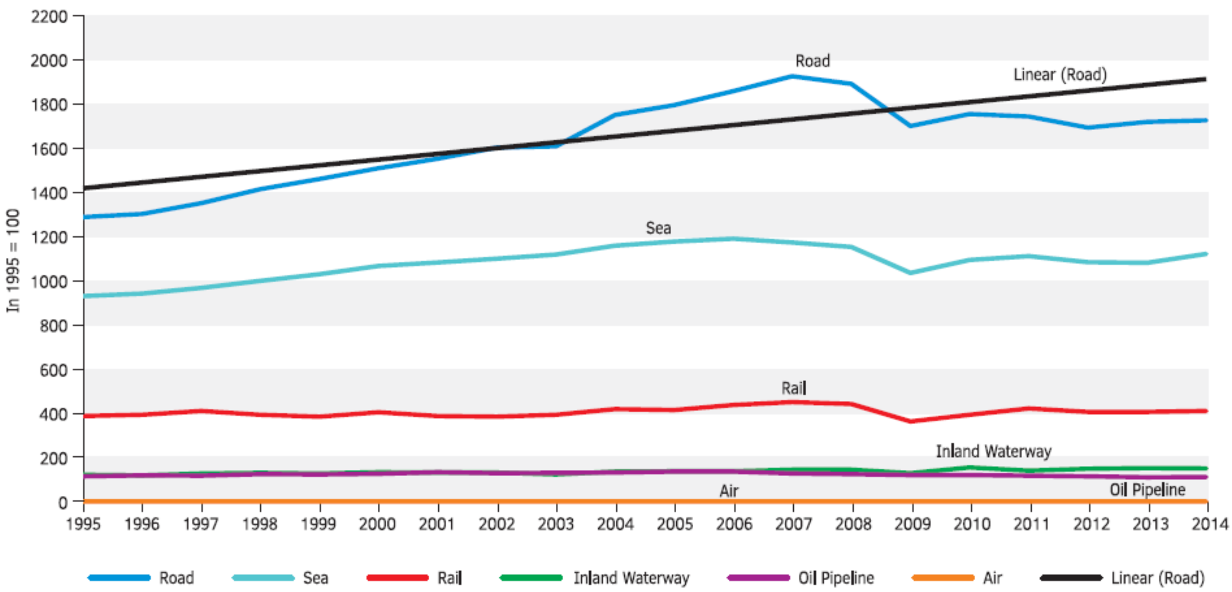

Figure 2.3: Size of the logistics market in Europe in the years 1995 to 2014 (million ton-kilometers).

Source: The Impact of Regulation of the Road Transport Sector on Entrepreneurship and Economic Growth in the European Union, Motor Transport Institute [13].

Table 2.1 shows a resume information of road freight transportation in the countries studied in this project. Portugal is a small country compared to the other 3, with a high dependency on road transportation. Looking at the other 3 countries more equal in size, Spain is the most competitive internally, having more than the triple of active companies than Germany. France has the biggest turnover and Germany, although with the least share of GDP, is the country with the most volume of carriage.

Table 2.1: Road freight transport data for 2013.

	Portugal	Spain	France	Germany
Gross Domestic Product (EUR millions)	170,269	1,025,634	2,115,256	2,826,240
Turnover (EUR millions)	4,796	29,996	43,679	39,194
Share of GDP	2.8%	2.9%	2.1%	1.4%
Volume of carriage (thousand of tons)	148,177	1,124,480	1,999,869	2,938,702
Staffing (thousands)	58.8	305.8	351.8	409.9
Number of companies	8,287	108,173	37,676	35,852

Source: The Impact of Regulation of the Road Transport Sector on Entrepreneurship and Economic Growth in the European Union, Motor Transport Institute [13].

2.2 Labour

When it comes to labour legislation, Europe is facing some problems in this sector. Germany and France have minimum wage laws (MiLoG law and Loi Macron law respective) that are preventing harmonized regulation. These laws impose that a driver operating in these countries have to be paid a minimum hourly rate fixed by these countries (€8.50/h for Germany and €9.76/h for France). This situation is allowing the possible grow of €1.4 billion/year in shadow economy with undeclared work [14]. Furthermore, companies that need to deliver or load at any of these countries will have added costs, not only the added hourly costs, but also high administrative burdens. Companies that do not deliver here but need to cross these countries are forced to rethink their strategy. Evaluating the possibility of going around, through alternative routes, that would imply different constraints (e.g.: quality of roads, more working hours due to longer routes, etc.), may have an effect on delivering times, among others, such as average speed. Many EU countries are disagreeing with these laws, invoking that these are in conflict with the freedom of the union internal market, for it disfigures the purpose of an Europe without internal borders.

Figure 2.4 shows the minimum hourly rate for Germany (green bar), and the rest of the EU-28, including Portugal (red bar). The discrepancy is high since many countries are too far from the reality of Germany and even more from France.

France for instance, requires companies to have a local representative for inspection reasons in addition to having locally documents of payrolls, order confirmations, and workers' contracts. Adding even more costs to outside companies.

Some benefit out of this situation: i) companies based in countries that have minimum wages higher than those of these countries are able to pay less, that is of course if their legislation allows it; ii) drivers crossing these countries have higher paychecks, for example, a driver departing from Spain with destination Poland, will most likely cross both France and Germany; and ideally, iii) Germany and France, by creating a certain equality within their borders, fighting social dumping. For these laws encourage local companies to remain in these countries and not go to near countries where labour is cheaper, and also companies already on near countries will not be an unfair competition when operating in France or Germany.

Every company operating in Germany or France, will have to pay these countries' minimum salary,

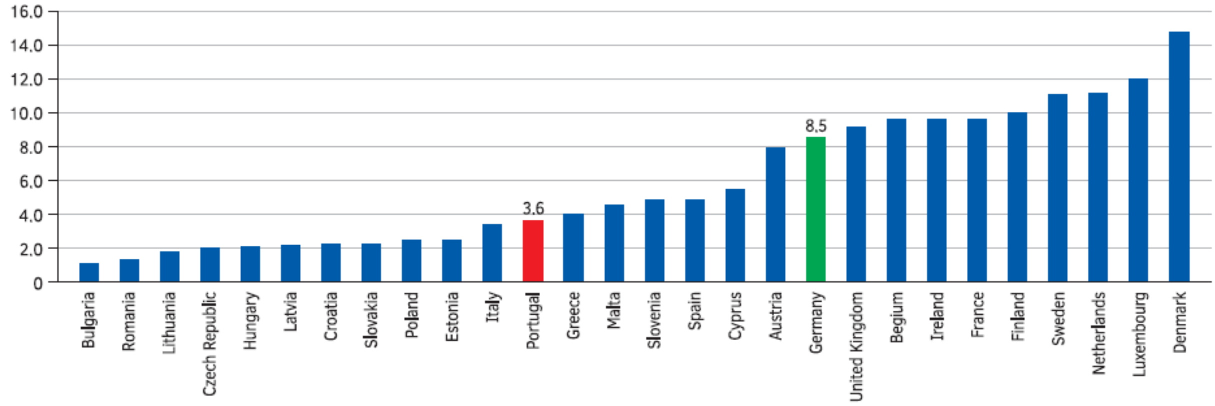

Figure 2.4: Minimum hourly rate (in EUR) compared to EU Member States in 2015.

Source: The Impact of Regulation of the Road Transport Sector on Entrepreneurship and Economic Growth in the European Union, Motor Transport Institute [13].

which can be too high considering a company's local economy. Not only that, but the drivers operating in these countries will most likely not live there, making their mandatory salaries disproportional to their economy, since these are not adjusted to their country's purchasing power. This situation might force companies to avoid crossing these countries.

Figure 2.5 is the same information of Figure 2.4, but adjusted to each country's purchasing power. The discrepancy is far inferior but still noticeable, especially in countries in the extremes. Denmark is still, too far from Bulgaria, in more than double.

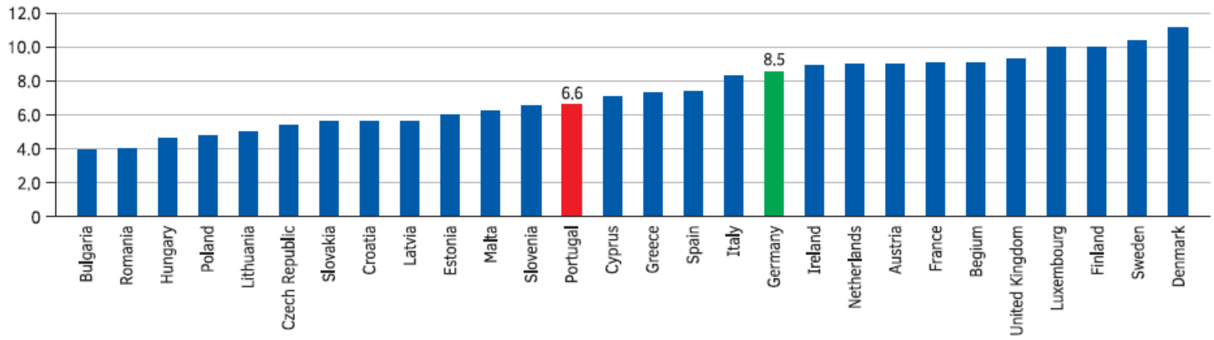

Figure 2.5: Minimum hourly rate (in EUR) compared to EU Member States, taking into account the purchasing power equivalent to the minimum hourly rate in Germany.

Source: The Impact of Regulation of the Road Transport Sector on Entrepreneurship and Economic Growth in the European Union, Motor Transport Institute [13].

A possible alternative could be a standard minimum wage equal for the whole union. However, this would require less deviation among countries' economies. For the countries in the extremes (Bulgaria and Denmark), it is extremely complicated to have the same minimum wage. On the other hand, uniforming the minimum wage for one sector alone, could be a first step in the direction of a stronger, cohesive, and more socially just Europe.

3: The Problem

The goal of a company in this business is to deliver (or pickup) an order to its destination, in time, and with the least possible costs.

A road freight transportation company has many expenses, be that with warehouses, fuel, tolls, trucks, among others. One particular expense is the expense with human resources, most, (if not all), companies have this expense. Salaries are one of the biggest costs a company can have, so it is on the interest of any company to minimize this expense. The road transportation is no different in this matter, what does differ is that drivers do not work (not only) from 8:00 to 18:00, they can work any hour of the day. Deciding when a driver works does have an impact on salaries' expenditure, furthermore, what driver delivers what order, also has an impact, because depending on where, what route to take, and how far the destination is, has an impact on what hours the driver drives. In the end the proper allocation of drivers has a considerable impact on a company's costs, being a problem worthy of study.

This project is focused only on the allocation of drivers ignoring everything else. Or in other words; considering everything else constant. For example, a company has a delivery to be made to a destination, (e.g.: Berlin), the truck is already loaded, and the route has already been determined. The almost all direct costs with this delivery have already been calculated, (fuel, tolls, etc.) what is left to calculate is who is going to drive this truck and what working schedules this person is going to follow. This is the focus of this work!

In road freight transportation besides salaries a company has to focus on a certain amount of rules obliged by local laws and when operating on foreign countries, also international laws. The following will be an exposure of the rules taken in consideration in this project.

3.1 Constraints

As it is expected, every activity has rules or limitations, in another word; constraints.

In this case, these constraints exist for several reasons: for fairness among competition; quality of labour for the drivers; security reasons, not only for the drivers, but also for everyone else sharing the roads with them; labour laws; etc..

Most of the constraints are permanent for the whole EU, but there are also special constraints regarding specific countries. Some are related to the country where the driver has its contract, some are applied only when present on that country.

Driving time and rest periods

The European Commission provides a common set of rules for driving and rest periods. These rules exist in order to guarantee fair competition, road safety, and drivers' good working conditions.

These rules are:

- Daily driving period shall not exceed 9 hours, with an exemption of twice a week when it can be extended to 10 hours.
- Total weekly driving time may not exceed 56 hours and the total fortnightly[1] driving time may not exceed 90 hours.
- Daily rest period shall be at least 11 hours, with an exception of going down to 9 hours maximum three times a week. Daily rest can be split into 3 hours rest followed by 9 hour rest to make a total of 12 hours daily rest
- Weekly rest is 45 continuous hours, which can be reduced every second week to 24 hours. Compensation arrangements apply for reduced weekly rest period. Weekly rest is to be taken after six days of working, except for coach drivers engaged in a single occasional service of international transport of passengers who may postpone their weekly rest period after 12 days in order to facilitate coach holidays.
- Breaks of at least 45 minutes (separable into 15 minutes followed by 30 minutes) should be taken after 4 ½ hours at the latest.

In this project not all rules will be applied and there are two main reasons for this.

Firstly, it is understandable that some of the rules have exceptions to allow drivers flexibility to deal with unpredictable events, such as traffic jams, accidents, flat tires, mechanical malfunctions, etc.. These are going to be ignored so that when needed, the driver has these exceptions stored as a last resort, to use outside of the driving plan in case of need. The second reason is to ease the programming, allowing a lighter program that works more efficiently. For that, is necessary that the results are produced within useful time, allowing a decision to be made.

[1] A period of 2 weeks/14 days.

Therefore the used rules are the following:

- **Daily driving period shall not exceed 9 hours.**

  E.g.: If a driver works on a Monday, from the 00:00 hours of that Monday until the 24:00 hours, the total time driving may not surpass 9 hours.

Table 3.1: Example of a driver's weekly schedule, with focus on 9h daily constraint[2].

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	...
00:00	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	...
01:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
02:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
03:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
04:00	0	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	...
05:00	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	...
06:00	0	1	0	1	1	0	0	1	0	1	1	0	0	0	1	1	...
07:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
08:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
09:00	1	0	1	1	1	0	1	0	1	1	1	0	0	0	1	1	...
10:00	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	...
11:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
12:00	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	...
13:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
14:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
15:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
16:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
17:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
19:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

$worked \leq 9h\ total\ per\ day$

[2] A "1" in a cell represents that 1 hour was driven from that cell's name to the next cell's name. E.g.: in the 1st column (Monday), the cell with name 07:00 has a "1", this means that the driver drove from the 07:00 hours to the 08:00 hours. A "0" on the other hand means that in that hour the driver was stopped. A better explanation will be provided in chapter 4.

- **Total fortnightly driving time may not exceed 90 hours.**

  E.g.: Similar to the 9 hours per day limit; a driver that has been driving from the 1st of the month, (from the 00:00 hours of Monday), until the 14th of the same month, (the 24:00 hours of Sunday), cannot have driven more than a total of 90 hours for that interval. The same from the 2nd of the month, (from the 00:00 hours of Tuesday), until the 15th of the same month, (the 24:00 hours of Monday). I.e.: In a period of 14 consecutive days a driver cannot exceed a total of 90 worked hours.

Table 3.2: Example of a driver's weekly schedule, with focus on 90h weekly constraint.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	...
00:00	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	...
01:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
02:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
03:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
04:00	0	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	...
05:00	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	...
06:00	0	1	0	1	1	0	0	1	0	1	1	0	0	0	1	1	...
07:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
08:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
09:00	1	0	1	1	1	0	1	0	1	1	1	0	0	0	1	1	...
10:00	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	...
11:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
12:00	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	...
13:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
14:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
15:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
16:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
17:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
19:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

$worked \leq 90h\ total\ per\ fortnight$

- **Daily rest period shall be of at least 11 uninterrupted hours.**

  The European Commission rules are not clear about the continuity of the daily rest period, it is implied, but there is some room for ambiguity. If the 11 hours rest wouldn't need to be necessarily continuous, that would mean that a driver could drive for 1 hour and then rest for 1 hour, then drive for 1 hour, rest 1 hour gain, and so on, for the whole day. There are other reasons to consider the continuity of this break that will be better explained later in this chapter.

Table 3.3: Example of a driver's weekly schedule, with focus on 11h daily rest constraint.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	...
00:00	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	...
01:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
02:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
03:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
04:00	0	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	...
05:00	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	...
06:00	0	1	0	1	1	0	0	1	0	1	1	0	0	0	1	1	...
07:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
08:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
09:00	1	0	1	1	1	0	1	0	1	1	1	0	0	0	1	1	...
10:00	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	...
11:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
12:00	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	...
13:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
14:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
15:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
16:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
17:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
19:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

$\geq 11h^{*} = (11h\ of\ rest\ per\ day + PT\ Law)$

- **Weekly rest is of at least 45 continuous hours.**

  E.g.: Between the 00:00 hours of Monday until the 24:00 hours of Sunday (of the same week), a driver has to have at least 1 interval of at least 45 uninterrupted resting hours. This applies to every set of 7 straight days. E.g.: From 00:00 hours of Monday to the 24:00 hours of Sunday, then the same to the 00:00 hours of Tuesday to the 24:00 hours of next Monday, and so on.

Table 3.4: Example of a driver's weekly schedule, with focus on 45h weekly rest constraint.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	...
00:00	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	...
01:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
02:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
03:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
04:00	0	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	...
05:00	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	...
06:00	0	1	0	1	1	0	0	1	0	1	1	0	0	0	1	1	...
07:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
08:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
09:00	1	0	1	1	1	0	1	0	1	1	1	0	0	0	1	1	...
10:00	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	...
11:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
12:00	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	...
13:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
14:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
15:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
16:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
17:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
19:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

$\geq 45h\ of\ continuous\ rest\ per\ week$

- **Breaks of at least 1 hour after a maximum of 5 straight hours of work.**

  E.g.: If a driver has started to work at 07:00 hours, by the 12:00 hours, if it hasn't had a break within this interval, it has to forcibly have one now. This break can be of just 1 hour, or more, but forcibly, it has to be of at least 1 hour.

Table 3.5: Example of a driver's weekly schedule, with focus on 5h continuous work constraint.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Monday	Tuesday	...
00:00	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	...
01:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
02:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
03:00	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	...
04:00	0	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	...
05:00	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	...
06:00	0	1	0	1	1	0	0	1	0	1	1	0	0	0	1	1	...
07:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
08:00	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	...
09:00	1	0	1	1	1	0	1	0	1	1	1	0	0	0	1	1	...
10:00	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	...
11:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
12:00	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	...
13:00	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	...
14:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
15:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
16:00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	...
17:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
19:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

$worked \leq 5h\ continuous\ work$

Since the program is designed to work with units of 1 hour, it isn't possible to work with 4 ½ hours or breaks of less than 1 hour. It is then rounded to a maximum of 5 hours and breaks of 1 hour. This is easily convertible by adjusting the program to work with fragments of hour instead of 1 hour units, (i.e.: ½ hour, ¼, or even intervals of a 1 minute). The trade-off is a heavier less efficient program.

The 56 hours per week limit is ignored due to the fact that it is impossible to exceed this without exceeding the 9 hour per day limit or the weekly rest of 45 continuous hours. If on a week, a driver would drive every day (9h per day), in the end of the week, it would have driven a total of 64 hours. But this would exceed the 45h rest limit. Assuming that a driver could drive the first 9 hours of a day (from 00:00 to 09:00), continuously, and the last 9 hours of the next day (15:00 to 24:00), it would only be a total of 30 continuous rest hours. To respect the 45h constraint, a driver can, at best, rest for a full day (it is impossible not to, since 45h of mandatory rest will always include a full day), continue to rest the next day for 21 hours (making it 45h), and then drive the last 3 hours of that day[3]. But this means that in a period of 2 days the driver has driven only 3 hours, adding this to the remaining 5 days of the week, sums to a total of 3 hours plus 5 times 9 hours, i.e.: 48 worked hours on that week.

Table 3.6: Example 1 of the 56h constraint redundancy.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	...
00:00	0	0	1	1	1	1	1	...
01:00	0	0	1	1	1	1	1	...
02:00	0	0	1	1	1	1	1	...
03:00	0	0	1	1	1	1	1	...
04:00	0	0	1	1	1	1	1	...
05:00	0	0	0	0	0	0	0	...
06:00	0	0	1	1	1	1	1	...
07:00	0	0	1	1	1	1	1	...
08:00	0	0	1	1	1	1	1	...
09:00	0	0	1	1	1	1	1	...
10:00	0	0	0	0	0	0	0	...
11:00	0	0	0	0	0	0	0	...
12:00	0	0	0	0	0	0	0	...
13:00	0	0	0	0	0	0	0	...
14:00	0	0	0	0	0	0	0	...
15:00	0	0	0	0	0	0	0	...
16:00	0	0	0	0	0	0	0	...
17:00	0	0	0	0	0	0	0	...
18:00	0	0	0	0	0	0	0	...
19:00	0	0	0	0	0	0	0	...
20:00	0	0	0	0	0	0	0	...
21:00	0	1	0	0	0	0	0	...
22:00	0	1	0	0	0	0	0	...
23:00	0	1	0	0	0	0	0	...

$\geq 45h\ of\ continuous\ rest\ per\ week$

[3] Alternatively, it could drive the first 3 hours of a day and the rest for the next 45 hours.

Another, and better way, would be to completely stop for a full day, 24h of rest, and split the remaining needed hours to complete the total of 45h rest, among the previously and following days. I.e.: a driver drives the first 9 hours of a day (even ignoring the mandatory rest of 1 hour per 5 continuous hours of work), rests the remaining 15 hours of that day. Rests completely the next day, (making it a total of 39 already rested hours). And on the third day, rests the missing 6 hours, and then works 9 hours. This would allow a driver to work 18 hours in a period of 3 days, while performing the mandatory 45h weekly rest. Adding the remaining 4 days of work, would mean a total of 9 worked hours in 6 days, that are 54 hours, less that 56 hours. Even if driving to a limit of 10 hour per day, 2 times a week, it wouldn't be exceeded. Which makes it redundant hence being ignored.

Table 3.7: Example 2 of the 56h constraint redundancy.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	...
00:00	1	0	0	1	1	1	1	...
01:00	1	0	0	1	1	1	1	...
02:00	1	0	0	1	1	1	1	...
03:00	1	0	0	1	1	1	1	...
04:00	1	0	0	1	1	1	1	...
05:00	1	0	0	0	0	0	0	...
06:00	1	0	1	1	1	1	1	...
07:00	1	0	1	1	1	1	1	...
08:00	1	0	1	1	1	1	1	...
09:00	0	0	1	1	1	1	1	...
10:00	0	0	1	0	0	0	0	...
11:00	0	0	0	0	0	0	0	...
12:00	0	0	1	0	0	0	0	...
13:00	0	0	1	0	0	0	0	...
14:00	0	0	1	0	0	0	0	...
15:00	0	0	1	0	0	0	0	...
16:00	0	0	0	0	0	0	0	...
17:00	0	0	0	0	0	0	0	...
18:00	0	0	0	0	0	0	0	...
19:00	0	0	0	0	0	0	0	...
20:00	0	0	0	0	0	0	0	...
21:00	0	0	0	0	0	0	0	...
22:00	0	0	0	0	0	0	0	...
23:00	0	0	0	0	0	0	0	...

$\geq 45h\ of\ continuous\ rest\ per\ week$

The remaining rules are ignored for the reasons already explained.

Portuguese Law

Table 3.8: Example of the exploit of the EU rules compared to the addition of the Portuguese Law.

	Driver 1	Driver 2
2017-01-01 03:00	0	1
2017-01-01 04:00	0	1
2017-01-01 05:00	0	1
2017-01-01 06:00	0	1
2017-01-01 07:00	0	1
2017-01-01 08:00	0	0
2017-01-01 09:00	0	1
2017-01-01 10:00	0	1
2017-01-01 11:00	0	1
2017-01-01 12:00	0	1
2017-01-01 13:00	0	0
2017-01-01 14:00	1	0
2017-01-01 15:00	1	0
2017-01-01 16:00	1	0
2017-01-01 17:00	1	0
2017-01-01 18:00	1	0
2017-01-01 19:00	0	0
2017-01-01 20:00	1	0
2017-01-01 21:00	1	0
2017-01-01 22:00	1	0
2017-01-01 23:00	1	0
2017-01-02 00:00	1	1
2017-01-02 01:00	0	1
2017-01-02 02:00	1	1
2017-01-02 03:00	1	1
2017-01-02 04:00	1	1
2017-01-02 05:00	1	0
2017-01-02 06:00	1	1
2017-01-02 07:00	0	1
2017-01-02 08:00	1	1
2017-01-02 09:00	1	1
2017-01-02 10:00	1	0

$\geq 11h^* = (11h\ of\ rest\ per\ day + PT\ Law)$

$worked\ \leq 5h\ continuous\ work$

Since in this project it is considered a Portuguese based company, the drivers are covered by Portuguese law, and it decrees that a driver (and any other worker) cannot work 2 consecutive driving periods (of 9 hours) without an interrupted rest period of 11 continuous hours, which in this case means that a driver can at best, work a total of 18 hours in a period of 31 hours. In Table 3.8 Driver 1 is exploiting this loophole, where Driver 2 is according to the Portuguese Law.

Given the European Union rules, a driver would never be able to drive 18 hours in the same day. However, a driver could drive two consecutive driving periods, if these periods would take place in the end of a day and on the immediate beginning of the next day. I.e.: The driver starts to work at 14:00 of Monday and finishes the day's shift at the 24:00 of the same day. Later at 01:00 of Tuesday, the driver starts to work again, finishing that day's shift at 11:00, making 18 hours of driving in less than 24 hours. This situation is feasible by European rules, it is a loop hole, since the rules mention daily driving, but not the day as a set of 24 continuous hours. It would be a tempting exploit of the rules, particularly regarding destinations within small distances. However, the Portuguese law forbids this, and to incorporate such law, the daily rest of 11 hours will be adjusted. Instead of considering periods of time as a singular civil day. A day, in this case, will be taken as a continuous sequence of 24 hours to make sure that a driver has 11 hours of uninterrupted rest after, or before, working a complete shift.

Locations

Some countries have their own specific driving constraints. Most of these constraints have to do with cargo, which is not a subject of study in this project. The constraints to have into account are related to weekends and holidays, namely in France and Germany. In France, it is not allowed to drive between the 22:00 of Saturday until the 22:00 of Sunday. The same applies to holidays and the day immediately before, e.i.: no driving allowed between the 22:00 of the day before a holiday and the 22:00 of the holiday.

Germany has similar constraints. It isn't allowed to transport goods through land between the 00:00 and 22:00 on Sundays and holidays.

Figure 3.1 gives a visual idea of when the driver is allowed to drive, where each circumference represents one of the used restrictions. Only when within all restrictions is the driver able to drive.

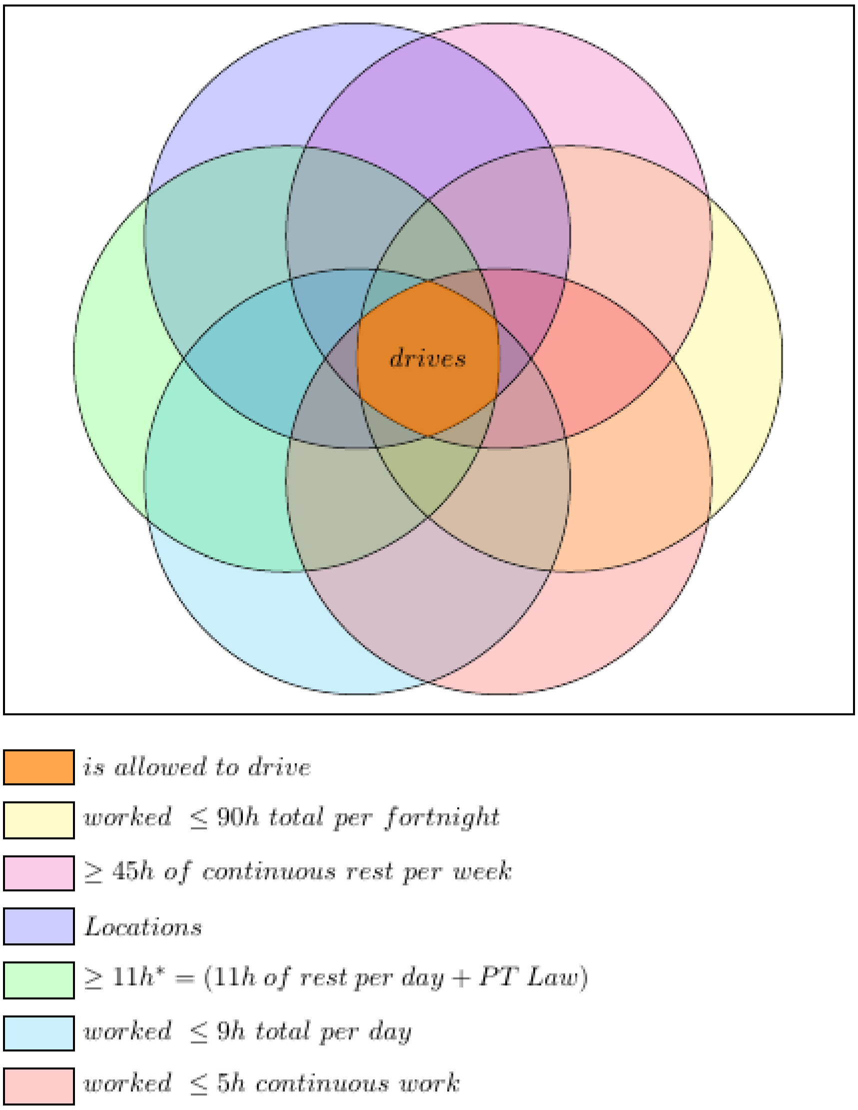

Figure 3.1: Representative diagram of constraints. A driver can drive if within the constraints.

(Disk's dimensions are not representative of the constraints' grasp).

3.2 Objectives

The goal of this project it to improve the allocation of drivers in order to have minimum global costs, satisfying the demand while reaching the destinations within an acceptable time window.

Time and costs are critical when allocating drivers. When it comes to costs the reasons are very straightforward; less direct costs mean more profit, be that by improving margins, practicing more competitive prices, a mix of both, or even using these savings in research and development. A more efficient company is a more competitive company. The proper allocation of drivers affects a company costs, not only deciding what driver must take the deliver, but also, when it is the best moment to depart, and what are the best hours to work, so that the restrains are respected but that the company does not endure in unnecessary costs. For example; it's 20:00 on a Monday and there is an order to be delivered to a 25 hours far destination. If a driver starts to drive now, it will drive the remaining 4 hours of Monday, drive 9 hours on Tuesday, plus 9 hours on Wednesday, and it reaches its destination on Thursday, at beast, at 3:00. Allocating the driver this way would incur in a direct cost of €133.75, (assuming that a driver is payed €5 per hour, and that in nightly hours it has an increase of 25%). If the company decides to wait and only let the driver start on the next morning at 8:00, the driver will reach its destination also on Thursday at the 15:00, for a total cost of €125. The driver can deliver the order in the same day with a difference of €8.75. It may not seem like much, but when scaling for a total number of delivers a driver does per year, and considering that a company has many drivers, a small amount such as this makes a big difference.

Time may not be so straightforward when it comes to its importance but it is also of great relevance in the freight transport business. Every client needs its orders in time, a company that takes longer than the competition to deliver, is a company that stays behind. It is important for a company's image, to be reliable and efficient. A company needs to deliver within what is the average market delivery time, but if it can do better, then it is more appealing for its customers. Allocating the right driver can make all the difference. Like in the example above, there is an order to be delivered to a 25 hours far destination, and there are two possible drivers. The 1^{st} driver is a driver that has a cumulative sum of 75 worked hours in the last 12 days, the 2^{nd} driver is a driver that for the same previously 12 days, has a cumulative sum of 65 worked hours. If the company chooses to allocate the 1^{st} driver, it will drive the first day for 9 hours, reaching a total of 84 hours in 13 days, and on the following day it will reach the limit of 90 hours per fortnight with 6 hours driven, being forced to stop for the rest of the day. Since it has 10 hours left to reach its destination and knowing that it can only drive a total of 9 hour per day, this means that the 1^{st} driver will take 4 days to reach this destination. The 2^{nd} driver on the other hand, can drive 9 hours on the first day, plus another 9 hours on the following day, and on the 3^{rd} day it will reach its destination after driving for 7 hours. In this example, allocating the right drive has a difference of a day and that can make a huge difference for the client.

Direct costs, and arriving costs (or indirect costs) are important individually, but they can also be improved together. In a pool of available drivers it is possible to have multiple drivers arriving at the destination in the same moment at different costs. The opposite is also possible, for the same cost, multiple drivers can reach the destination in different moments. Reaching the destination as soon as possible with the minimum possible cost is the ideally situation, when this is not possible it is required to define what weights the most, the arriving moment or the cost? Is it worth allocating a driver that arrives an hour late if that means saving €20? Arriving a full day sooner is worth allocating a driver that will have an higher cost of €10?

The all drivers' allocation or even just a driver's schedule has a large amount of feasible solutions that respect these constraints. Optimizing a driver's schedule resorting to operations research, in specific, integer programming, would take a considerable amount of computer processing, which translates into many hours of calculations until the optimum solution would be reached. The farther the destination the more time would take to achieve the best solution. Allocating the whole drivers' fleet this way would just increase this situation exponentially.

To avoid this, it was used a methodology focused on the goal of making a driver reach its destination as fast as possible, respecting the existing constraints. Once this information is calculated for every driver on the fleet, it was then optimized the allocation using now operations research.

In chapter 4 it will be explain in more depth the methodology used to achieve these goals.

4: Methodology

A program was developed in order to help in the decision of allocating drivers to the different destinations. The goal of this program was to create different schedules for all the available drivers. Where a driver departs as soon as it is able, driving every day it can, and as soon as it can on each day, until the destination is reached. Then the program calculates the same but departing 1 hour later, and then 1 hour more, and so on, for as many times (hours) as the manager decides it is important to analyze. For example, a driver can depart as soon as 03:00, the second schedule will be with the driver departing at 04:00, and the third at 05:00, and so on. Then this process is repeated for every destination and every driver.

After all this information has been calculated, begins the allocation optimization. Where it is calculated the cost per each schedule, both direct costs with drivers hourly rates, and indirect costs with arriving moments, and then according to the demand, it is optimized the best global allocation that satisfies this demand with minimal costs.

4.1 Specifications

All software used in this project is open source, with exception of Microsoft's Excel. These are; R, R-studio, QGIS and LibreOffice. The date format used is the indicated by the ISO norm 8601. The average speed and the routes/roads taken to reach the destination are consider as inputs. The timezone used is the western European time zone UTC+00:00. The difference in time zones is ignored for what is important is how long the driver is driving, regardless of what the local clocks present.

The departing city is Lisbon and there will be 3 possible destinations; Madrid, Paris and Berlin. Once a driver, allocated to a destination, reaches it, it will immediately start to travel back. It will be ignored any time spent with loading and unloading the trailer. It is assumed that the trailers are timely loaded for the drivers and that a driver, upon arriving, switches trailers and keeps on moving (if eligible to do so). This is ignored for several reasons. Usually the drivers are not the ones who load or unload the trailers, this work is generally accomplished by the warehouse workers. Another reason, and most importantly, is that it is not wise to define a rule that forces a driver to remain in a destination for an extra hour, since it is not unusual for a driver to have to be stopped due to its own constraints. Therefore there would be many occasions where the driver would have its trailer unloaded and reloaded, ready for the trip back while resting. But in other occasions it would have to be forced to stay idle, despite the fact that it could be on the move already. It would be possible to solve this without much problem, but that would not make the program necessarily more realistic, especially considering that a switch of trailers is quite realistic in a company that has a high frequency of shipments. This can, however, be added to the heuristic later if eventually wanted or needed.

Cabotage[1] was ignored in this project. Adding this would increase the level of complexity of the problem, requiring that the program would have to calculate more information. For example, a driver delivers an order from Lisbon to Berlin and upon returning brings a delivery from Berlin to Paris, and then another delivery from Paris to Madrid. This would require an advance adjustment of dates, for pickups and dropouts, and an adjustment to have a driver in a compatible route when those delivers exist. It is possible that the program can be adjust to this from of activity, although that surpass the environment of this project, and for now, it is ignored.

4.2 Inputs

For the program to function properly it needs to adjust to the company's reality, to do so, some inputs are required. This is a situation of continuity. There is a recent pass that has to be taken into consideration, namely, the driver's driving history.

Other information that the program needs beforehand are the lists of holidays for each of the countries where the company operates. These are of significant impact, needed for the driving constraints, and for the calculation of costs.

Another indispensable information are the destinations' distances. The program needs to know where to send the drivers, or in the program's understanding, how many hours the driver has to drive to reach its destination.

On the financial side of things, the program needs to know how are the costs calculated. I.e.: what is the hourly rate for a driver to drive on a particular hour, in order to present the total cost per driver.

A final input necessary is the interval of days for the program to work with. The program needs to know where is working, in time, so that it can take into consideration the proper driver's history, the holidays, and the interval limit of time to work.

Holidays

The Holidays' information is something that can be updated only once - in the beginning of every year. Most holidays are stationary, and although they are all predictable, some are harder to calculate, as it is the case with Easter and other religious holidays. Not to mention that for political, economical or social reasons, holidays may be added or excluded as such.

The holidays to have into consideration are only the holidays of countries where the company operates, and even so, not all countries' holidays may have to be considered. Many countries have regional holidays that may not affect the company, and other countries' holidays don't have any impact at all, such is the case of Spain.

As mentioned in chapter 3, France and Germany have driving constraints that are associated to these countries' holidays. Without knowing when these holidays occur, the program would be at fault, creating inadequate schedules, disrespecting these countries' rules, leaving the company with unnecessary issues, with the possibility of incurring in unnecessary expenses with fines for example.

[1] Cabotage is the transport of goods in a country by a transport operator from another country.

Another reason why holidays are an important input has to do with the company's head quarters being in Portugal, and the drivers being covered by Portuguese law. This means that driver's variables pays are influenced by holidays in Portugal.

Costs

In order to accurately calculate a driver's schedules cost, it is necessary to know how much a driver's hourly rate is. To that, it is also required to add the information regarding night hourly rates, holiday rates and weekend rates. For this project, the following rates were required: the Portuguese Labour Law rates; and eventually other rates could be needed, such as, different basic hourly rates per drive. E.g.: A driver that has been with the company for several years might have a higher hourly rate than others.

The costs with trucks, regarding maintenance, fuel, tolls or any other expenses are ignored. It is assumed that these costs are constant with the destination. The costs of shipment from Lisbon to Berlin are greater than shipping from Lisbon to Madrid particularly with fuel and tolls, but that is not taken in consideration, since the destinations will not be competing among themselves. The focus is on satisfying every destination's need.

It will be assumed that the drivers have a fixed wage and that they are paid an extra over worked hours. By default a driver is payed 5 units per worked hour. I.e.: if a driver is on route but it is idle, it it won't get paid. If on the other hand, it is driving, then it receives, (by default) 5 units per hour. As previously mentioned on chapter 3, the drivers are covered by Portuguese law. This states that the worked hours on particular cases have a percentage added to the default value. When driving at nigh hours, drivers get an extra 25% over the standard value, making it a 6.25 units. A night hour is an hour that is contained in the interval from 22:00 of a day to the 07:00 of the following day, inclusive. On the weekends drivers get payed an extra 50% (7.50 units). And on Holidays an 75% extra. All of this is considered in Portuguese time zone, as if the driver is in Portuguese territory. E.g.: a driver is driving in Germany, and the local time is 22:00 (UTC+01:00), the driver will not be paid as a night hour, since in Portugal it is 21:00 (UTC+00:00). On the other hand, if a driver is driving at 07:00 local time and it is still 06:00 in Portugal, then the driver is paid as a night hour. The same is applied to weekends and holidays. These add values are non cumulative, prevailing the highest values. E.g.: A driver is working on a holiday, which happens to be on a weekend and at night, then the driver only gets paid the extra 75%, not a cumulative 150% (25% + 50% + 75%).

Arriving to a destination at different possible times has to be convertible to a mensurable meaning. In order to do so, it is considered the soonest moment of arrival to a destination as a 0 cost. Arriving 1 hour later, will be considered as 1 unit of costs, 2 hours later, 2 units of costs, an so on. E.g.: a driver can arrive the soonest to Paris at 2017-01-31 05:00, this will be considered to cost 0 units. This is the standard arriving moment for this destination. An alternative arriving moment, such as 2017-02-02 00:00 will be converted to 43 units of cost, because the driver arrives 43 hours later.

The MiLoG and Loi Macron laws are ignored, for it was not clear how this laws are applied in combination with the Portuguese Labour Law. Te loss of this information does not have a high impact on the goal of this project, and it can easily be added to the costs' calculation in a future version of the program.

Different solutions would possible be presented, but the demand would still need to be satisfied, and the countries in question would not be avoided in any way since routes are not a subject of study here. Regardless, the program is able to incorporate these laws if wanted.

Destinations

In order for the program to calculate the schedule for a driver, it needs to know what this driver's destination is. For the program, a destination is a total number of working hours, so, it is then necessary to tell the program how many hours it takes to reach each destination.

To calculate a destination's length in hours, it was taken into consideration the average speed of 80 kilometers per hour. Therefore, a distance of a 1,000 km is interpreted as 13 driving hours.

Routes are not taken into account because it is considered that there aren't many significant available options, i.e.: options that actually have an impact on this type of project. Obviously there are many alternative routes that reach the same destination, but they are hardly ever going to avoid a main freeway that consumes most of the traveling time. When it comes to national roads there are plenty of alternatives, and many of which might be important due to several reasons, such as road traffic depending on when they are traveled; constructions; road repairs; etc.. That, however, is beyond the scope of this project. It is just accepted that the routes are the best in a long term and that on average, regardless of the route, a driver takes a given amount of time to reach its destination and it is based on that amount that the program functions.

Since for this project those statistics were not available, it was calculated the time a driver needs to reach its destination by dividing the total distance in kilometers by the average speed mentioned.

Driver's history

To know if a driver is eligible to drive, respecting the mentioned constraints, it is indispensable to have the information regarding drivers' history of previously worked hours. The farthest this history has to go is the previously fortnight[2]. Farther information is not necessary, but this is, in order to respect constraints like: the 90 hours limit per fortnight, the 45 hours rest per week, and, the 11 hours rest per daily shift. Even if a driver is technically new and has no driving history it has to have this information available. In this case it just so happens to be a fortnight of comply rest. Same goes for drivers returning from holidays. The program will not work without this information.

Time window

Finally and more trivially, the program requires a time window to be defined. The starting day (and hour) must be equal to the first day of the inputted drivers' history, and the ending day must be far enough for the program to work. I.e.: the last fortnight plus enough days for the driver to return from its destination, with days to spare. Having more days than exactly needed will not have significant influence on the program's performance, since the program stops as soon as the driver has returned. On the other hand, having insufficient days will prevent the program from working properly.

This input is not an important part of the problem as much as a result of the design of the program. The program can be improved in order not to be dependable of the user to calculate this information, since it could probably just estimate the length of this time window, by looking at the available drivers' history ending date, and then adding more days/hours as it would be needed during the calculation of the schedules.

[2] To be precise only the previously 13 days are actually needed, since the 14th day will be the currently day being analyzed by the program.

This, however, would require more developing time and since it didn't have much impact on the end result, it was ignored. On the other hand, this allows testing the program for different time windows and other aspects, by giving the user more control.

4.3 Heuristic

For a driver's schedule for only 1 day, from the 00:00 to the 24:00 hours, assuming that this driver drives a total of 9 hours, and that it does not drive more than 5 straight hours, there are over a 1.000.000 feasible solutions[3]. To reach THE optimum solution for minimum costs for this particular day, it would require the use of integer programming, which is usually a slow process due to its characteristics. The more feasible solutions and more constraints, the slower the optimization is, in the previous example it was ignored the fact that a driver can drive less than 9 hours in a day, and all other constraints used in this project[4], meaning that the number of admissible solutions would be even bigger. To optimize (minimizing costs) the hole allocation of every admissible driver, satisfying the demand and respecting every constrain while using integer programming would take a long time. If this process would start and at the same time a driver would depart for a destination, this driver would have return before the process would be concluded. To avoid this time consuming, it was built an heuristic that finds A solution, a called "greedy heuristic", that is designed to make the driver drive every time it is eligible until the destination is reached, (and then return), ignoring costs. This is calculated for several different departing moments per driver, and only after all drivers have all this calculated is then used integer programming for the optimization for minimum costs, with a considerable less amount of data to be processed in the optimization, gaining time at the expense of missing the optimum solution, but finding a solution that is useful. The used heuristic is a simple one that can be improved, but the more rules the heuristic has the more likely it is to be slower. However, it is possible to have more rules that compensate their existence by avoiding unnecessary calculations that translate in better overall performance.

By default the driver is stopped (using binary; 0). The program analyses one hour at a time. It asks itself "can I drive in this hour?", and if no constraints forbid it, then the driver drives that given hour[5] (in binary; 1). And so on, until a constraint forbids it, or the destination is reached. It is also designed so it doesn't waste too much computation with a period of time that isn't going to be available soon. I.e.: if the driver has just finished a daily period of driving, there is no point analyzing the following hours, because it is a resting period, so it jumps to the next available time when it might be able to drive, which in this case would be the following day. This way the program becomes more efficient, allowing faster results. In a small project with small variables, speed may not be a concern, but when applying large quantities of real data, a slow program can become unusable by not providing a result in time for a decision to be made.

In Table 4.1 there is an example of a driver's schedule. The dates and time on the left column indicate when a driver starts (or not, 0) to drive, and it drives until next hour. E.g.: in the tenth row this driver is schedule to drive the 09:00 hour of the 1st day of January, meaning that this driver will be working when the clock marks 09:00 and finishes when the clock marks 10:00, and at 10:01 it will be stopped until the next day.

[3] 1.295.128 to be precise.
[4] As mentioned in chapter 3.
[5] Figure 3.1.

Table 4.1: Example of a driver's schedule.

2017-01-01 00:00	1
2017-01-01 01:00	1
2017-01-01 02:00	1
2017-01-01 03:00	1
2017-01-01 04:00	1
2017-01-01 05:00	0
2017-01-01 06:00	1
2017-01-01 07:00	1
2017-01-01 08:00	1
2017-01-01 09:00	1
2017-01-01 10:00	0
2017-01-01 11:00	0
2017-01-01 12:00	0
2017-01-01 13:00	0
2017-01-01 14:00	0
2017-01-01 15:00	0
2017-01-01 16:00	0
2017-01-01 17:00	0
2017-01-01 18:00	0
2017-01-01 19:00	0
2017-01-01 20:00	0
2017-01-01 21:00	0
2017-01-01 22:00	0
2017-01-01 23:00	0
2017-01-02 00:00	1
2017-01-02 01:00	1
2017-01-02 02:00	1
2017-01-02 03:00	1
2017-01-02 04:00	1
2017-01-02 05:00	0
2017-01-02 06:00	1

The heuristic keeps working like this until the total sum of worked hours is equal to the necessary hours for the driver to reach its destination and return. Then it does the same but on another schedule. The first schedule will start as soon as the driver is able to start driving, or in other words, ready to depart. The following schedule will start 1 hour later than the previous, and so on, until the total number of schedules asked to be calculated will be completed. Then it moves on to the next driver, repeating the process until all schedules of all drivers are completed for all destinations.

Once this information is processed, a table like Table 4.2[6] is created with each driver and their own respective (different) schedules. It shows the days and hours where the driver is going to be in; the

[6]Due to its size, Table 4.2 is available as an annex.

departing city (Lisbon), each nearest frontier city (e.g.: Badajoz), the destination city (e.g.: Madrid), and finally the returning date. It also presents the total traveling time to reach the destination city, the total time to be back at the company, and the total cost of the trip. This table is then exported to a csv file for ease of read.

This information is also exported to be used on QGIS as a visual aid with a map of the visited countries and moments where the driver is on certain cities. In Figure 4.1 driver 1 with schedule 1 is departing from Lisbon on 2017-01-15 00:00, arriving at Guarda on 2017-01-15 04:00, reaching San Sebastian on 2017-01-16 03:00, passing through Saarbrucken on 2017-01-17 10:00, and arriving to the destination Berlin on 2017-01-21 03:00.

Figure 4.1: Example of the map from Lisbon to Berlin, with driver 1, schedule 1 information.

After producing every file per destination two global matrices are then created, products of a combination of every destination file. These files will have the costs per destination. One of these files (Table 4.3) will have the costs of worked hours and the other (Table 4.4) will have the costs per arriving moment in which the later one must be converted to tangible costs (Table 4.5). Then both of these files will be merged together with the importance of each file determined by the manager or the company. E.g.: the company determines that the costs of driven hours have 90% impact and the arriving time has only 10% of impact. This unbalance (later defined as alpha) is defined here when both files are combined, and it is completely up to the company or manager to decide. This information is needed to determine the best combination of drivers sent to fully satisfy the destinations' needs.

Table 4.3: Example of the Complete Resume Matrix Hourly costs for 3 destinations.

Driver type	Schedule	Madrid	Paris	Berlin
5	22	97.50	270.00	433.75
5	23	97.50	271.25	435.00
5	24	96.25	270.00	433.75
6	1	97.50	261.25	438.75
6	2	95.00	260.00	438.75
6	3	93.75	257.50	435.00
6	4	93.75	256.25	432.50
6	5	92.50	255.00	431.25
6	6	91.25	258.75	438.75
6	7	90.00	257.50	437.50
6	8	88.75	256.25	436.25
6	9	87.50	255.00	435.00
6	10	87.50	255.00	435.00
6	11	87.50	255.00	435.00
6	12	87.50	255.00	435.00
6	13	87.50	255.00	435.00
6	14	87.50	255.00	435.00
6	15	88.75	256.25	436.25
6	16	91.25	258.75	437.50
6	17	92.50	260.00	438.75
6	18	93.75	261.25	440.00
6	19	93.75	261.25	440.00
6	20	95.00	263.75	442.50
6	21	96.25	262.50	441.25
6	22	97.50	265.00	442.50
6	23	97.50	266.25	432.50
6	24	97.50	267.50	435.00
7	1	95.00	260.00	438.75
7	2	93.75	257.50	435.00
7	3	93.75	256.25	435.00
7	4	92.50	255.00	433.75
7	5	91.25	258.75	438.75

Table 4.4: Example of the Complete Resume Matrix Arriving moments for 3 destinations.

Driver type	Schedule	Madrid	Paris	Berlin
5	22	2017-01-18 06:00	2017-01-20 03:00	2017-01-24 08:00
5	23	2017-01-18 07:00	2017-01-20 03:00	2017-01-24 07:00
5	24	2017-01-18 08:00	2017-01-20 03:00	2017-01-24 07:00
6	1	2017-01-17 08:00	2017-01-19 03:00	2017-01-23 00:00
6	2	2017-01-17 09:00	2017-01-19 04:00	2017-01-23 01:00
6	3	2017-01-17 10:00	2017-01-19 04:00	2017-01-23 00:00
6	4	2017-01-17 11:00	2017-01-19 04:00	2017-01-22 23:00
6	5	2017-01-17 12:00	2017-01-19 04:00	2017-01-20 10:00
6	6	2017-01-17 13:00	2017-01-20 04:00	2017-01-23 08:00
6	7	2017-01-17 14:00	2017-01-20 04:00	2017-01-23 08:00
6	8	2017-01-17 15:00	2017-01-20 04:00	2017-01-23 08:00
6	9	2017-01-17 16:00	2017-01-20 04:00	2017-01-23 08:00
6	10	2017-01-17 17:00	2017-01-20 04:00	2017-01-23 08:00
6	11	2017-01-17 18:00	2017-01-20 04:00	2017-01-23 08:00
6	12	2017-01-17 19:00	2017-01-20 04:00	2017-01-23 08:00
6	13	2017-01-17 20:00	2017-01-20 04:00	2017-01-23 08:00
6	14	2017-01-17 21:00	2017-01-20 04:00	2017-01-23 08:00
6	15	2017-01-17 22:00	2017-01-20 04:00	2017-01-23 08:00
6	16	2017-01-17 23:00	2017-01-20 03:00	2017-01-23 07:00
6	17	2017-01-18 00:00	2017-01-20 03:00	2017-01-23 07:00
6	18	2017-01-18 01:00	2017-01-20 03:00	2017-01-23 07:00
6	19	2017-01-18 02:00	2017-01-20 03:00	2017-01-23 07:00
6	20	2017-01-18 03:00	2017-01-20 03:00	2017-01-23 06:00
6	21	2017-01-18 04:00	2017-01-20 03:00	2017-01-23 07:00
6	22	2017-01-18 05:00	2017-01-20 03:00	2017-01-23 05:00
6	23	2017-01-18 06:00	2017-01-20 03:00	2017-01-24 04:00
6	24	2017-01-18 07:00	2017-01-20 03:00	2017-01-24 03:00
7	1	2017-01-17 09:00	2017-01-19 04:00	2017-01-23 01:00
7	2	2017-01-17 10:00	2017-01-19 04:00	2017-01-23 00:00
7	3	2017-01-17 11:00	2017-01-19 04:00	2017-01-22 23:00
7	4	2017-01-17 12:00	2017-01-19 04:00	2017-01-20 10:00
7	5	2017-01-17 13:00	2017-01-20 04:00	2017-01-23 08:00

Table 4.5: Example of the Complete Resume Matrix Arriving moments adjusted to costs, for 3 destinations.

Driver type	Schedule	Madrid	Paris	Berlin
5	22	70	92	173
5	23	71	92	172
5	24	72	92	172
6	1	48	68	141
6	2	49	69	142
6	3	50	69	141
6	4	51	69	140
6	5	52	69	79
6	6	53	93	149
6	7	54	93	149
6	8	55	93	149
6	9	56	93	149
6	10	57	93	149
6	11	58	93	149
6	12	59	93	149
6	13	60	93	149
6	14	61	93	149
6	15	62	93	149
6	16	63	92	148
6	17	64	92	148
6	18	65	92	148
6	19	66	92	148
6	20	67	92	147
6	21	68	92	148
6	22	69	92	146
6	23	70	92	169
6	24	71	92	168
7	1	49	69	142
7	2	50	69	141
7	3	51	69	140
7	4	52	69	79
7	5	53	93	149

To finish, resorting to linear programming using an R package called ”Rglpk”, it will be determined what set of drivers will be traveling to what destination and on what schedule, minimizing the global costs.

4.4 Types of driver

When dealing with a large enough amount of drivers, many will have equal schedules, with equal arriving moments and costs. It is wise to typify the drivers, for both speed and efficiency.

Suppose a pool of 100 drivers, 3 destinations, and a desire to analyze 24 different schedules, the heuristic will run an amount of 7,200 times. If out of those 100 drivers, 70 are replications of the other 30, the program will only need to run 2,160 times, which will result in a gain of 70% efficiency. However, there isn't a constant percentage of gain out off n drivers. The suspicion is that for a constant number of destinations, the greater the number of drivers the greater the efficiency gain by typifying the drivers. Something that is useful, since the greater the number of drivers, the longer the heuristic needs to run. On the other hand, the farthest the destination, the hardest it is to typify drivers. Unfortunately, this goes beyond the objective of this project and it will not be studied in more depth this time.

The best way to determine if drivers are similar is to run the program for the farthest destination and in the shortest schedule, the most demanding criteria. This will be the best way to determine if a driver is most likely to be limited by a constraint, thus differentiating itself from the others. Two similar drivers for shorter destinations can be different when on long routes, due to wider constraints that would not manifest in shorter routes. A later schedule means that the drivers have time to rest, making them less subjective to constraints, whereas the soonest schedule will have bigger odds to meat a constraint.

A unique driver must differ from other on at least 1 of the following; departing moment, arriving to destination moment, returning moment, or total cost. A driver is then to be consider redundant, if there is a driver that equals this one in every one of these criteria.

Table 4.6 represents an example of equivalent drivers. They all work on the same day, and all have the same total hourly cost. Driver 1 and driver 2 are completely equal, so there is no need to recalculate everything for both drivers, since both are exactly equal, this way driver 1 and driver 2 can be typified has driver type 1. Driver 3 is different from drivers 1 and 2 since it has a break sooner, but it costs the same, departs at the same time, and arrives at the same time, (for this example it is ignored the returning moment), so driver 3 will too be considered as driver type 1, since it is equal in all criteria to drivers 1 and 2. Driver 4 departs at the same time as a driver of type 1, costs the same, but arrives in a different moment, so driver 4 will now be a different type of driver, it will then be driver type 2. Driver 5 is different from drivers type 1 and 2 because it departs at a different moment than both of the existent types, so it will be a new type of driver, driver type 3. Finally, driver 6 although different from all other drivers it is equal to drivers of type 2 since it costs the same, and arrives and departs at the same moments as this type of driver, so it will be considered as another driver of type 2. In the end, from the 6 drivers, were typified 3 types of drivers. Where driver type 1 has 3 drivers that are drivers 1, 2 and 3. Driver type 2 has 2 drivers that are driver 4 and 6, and driver type 3 has 1 driver that is driver 5. In this example, out of a fleet of 6 drivers the program will only need to calculate schedules for 3 drivers, half the initial amount of calculations needed.

Table 4.6: Example of equivalent drivers.

	Driver 1	Driver 2	Driver 3	Driver 4	Driver 5	Driver 6
2017-02-01 00:00	0	0	0	0	0	0
2017-02-01 01:00	0	0	0	0	0	0
2017-02-01 02:00	0	0	0	0	0	0
2017-02-01 03:00	0	0	0	0	0	0
2017-02-01 04:00	0	0	0	0	0	0
2017-02-01 05:00	0	0	0	0	0	0
2017-02-01 06:00	0	0	0	0	0	0
2017-02-01 07:00	0	0	0	0	0	0
2017-02-01 08:00	1	1	1	1	0	1
2017-02-01 09:00	1	1	1	1	1	1
2017-02-01 10:00	1	1	1	1	1	1
2017-02-01 11:00	1	1	1	1	1	0
2017-02-01 12:00	1	1	0	0	1	1
2017-02-01 13:00	0	0	1	0	1	1
2017-02-01 14:00	1	1	1	1	0	1
2017-02-01 15:00	1	1	1	1	1	0
2017-02-01 16:00	1	1	1	1	1	1
2017-02-01 17:00	1	1	1	1	1	1
2017-02-01 18:00	0	0	0	1	1	1
2017-02-01 19:00	0	0	0	0	0	0
2017-02-01 20:00	0	0	0	0	0	0
2017-02-01 21:00	0	0	0	0	0	0
2017-02-01 22:00	0	0	0	0	0	0
2017-02-01 23:00	0	0	0	0	0	0
Total direct cost	45	45	45	45	45	45

4.5 Co-op drivers

Many companies use teams of drivers in a single deliver, this allows arriving to the destination sooner, and also with more security, since the resting driver can always make sure the active driver is well and in good conditions to drive, and not over-burned. But of course there are trade-offs, such as, more possible expenses, conflicts between the drivers due to lack of team spirit or territorial behavior, among possible others.

This will be considered in the study because it is important to identify what destinations are worth to be assigned teams of two drivers rather than one, when it comes to minimizing time spent delivering. But most of all, to make the program able to give this information to the manager as it be required or wanted.

When adding co-op drivers to the analysis, a pool of n drivers increases to $n(n+1)$ sets of drivers. The original n drivers (driving alone), plus every n combination of 2, $({}^{n}C_2)$, times 2, (since order matters). The driver who starts to drive first has to be eligible to do so, while the second driver can rest and it is only needed to be eligible to driver later. By changing the order of drivers, results may vary, so it is needed to be added to the study, thus $2 *^n C_2$. If typification of drivers is used, to the original n type of drivers another n type of drivers must be added, (since each type of driver will very likely represent multiple drivers). These are the repeated pairs, such as type of driver 1 with itself. E.g.: 1 with 1, 2 with 2, and so on. Making it a $2 * (n +^n C_2)$, that with a little math manipulation simplifies to $n * (n+1)$. Without using driver's typification, therefore no doubles, i.e.: 1 with 1, 2 with 2, etc.. Then it is n^2 instead of $n(n+1)$. If a company has 6 different drivers, (no typification), they can all driver alone (6), plus every unique combination among the 6 $\left({}^{6}C_2 = 15\right)$, times 2 (because the combination driver 6 with driver 2 is different from the combination driver 2 with driver 6), making it a total of 36 $\left(6^2\right)$. If a company has 6 types of drivers, then each one of these 6 types of drivers can drive alone (6), plus every unique combination among the 6 $\left({}^{6}C_2 = 15\right)$, times 2, plus every double (driver type 1 with driver type 1, driver type 2 with driver type 2, and so on) that is 6, so 6 alone + $2 *^6 C_2$ + 6 doubles, making it 42, $6(6+1)$.

Looking back to the argument for the typification of drivers and maintaining the same number of drivers, schedules and destinations; adding co-op drivers to the mix, from 100, there are now 10,100 drivers, times 24 schedules, times 3 destinations, equals 727,200 runs of the program. By using typification 10,100 are reduced to 930 $(30\,(30+1))$, times 3 destinations, times 24 different schedules, equals having the heuristic running 66,960 times. This is now, not a 70% increase in efficiency but rather, a more than a 90% improvement. The heuristic is adapted to calculate every destination for the co-op system, and then adds this information to the singular driver system. After this, the optimization takes place, now with some special attention to the allocation of a pair of drivers, that costs 2 drivers, and a singular driver that only costs 1, to satisfy a destination's singular demand.

5: Results

To prove the concept of this project it was taken as an input, a pool of 51 drivers, where the 1st driver was a new driver with no driving record. The remaining 50 drivers had a driving history that was a result of previous tests conducted to the heuristic and creative ways to simulate records. This history of driven hours was not based on real existing drivers but several tests' results made to the program that somehow make sense as a driving record. The focus was that the drivers' history was meaningful and diverse.

This simulation was started at 00:00 of the 15th of January, 2017. Consequently, the drivers history was from the 00:00 of the 1st of January, 2017, up to the 23:00 of the 14th of January, 2017.

The considered destinations were 3: Madrid, with a distance of 8 hours; Paris, with a distance of 22 hours, and finally, Berlin, with a distance of 36 hours. It was also taken in consideration the returning trip, meaning, 2 times the original destination distance. E.g.: Going to Paris takes 22 hours, plus an extra 22 hours to return. And there were calculated 24 different schedules per driver and co-op drivers. As a result, the 24 schedules gave a spectrum view of the leaving impact at any of the 24 different hours of a day.

Table 5.1[1] is an example of the 24 hours of the first day, of the first 31 drivers history, where each column represents a different driver. The remaining information was omitted due to the size of the table. This information was then processed in order to be shrunk to a types of driver table, that would omit redundant information allowing the faster and more efficient performance of the program.

5.1 Types of drivers

Firstly a filtration by typifying drivers was conducted. Asking the heuristic to calculate the 1st schedule, which means, departing as soon as possible, to the farthest destination (Berlin, 36 hours), and back, (another 36 hours), making it a total of 72 hours. The program took 5.700763 mins to calculate this information in a Linux machine fully dedicated to this operation.

In Figure 5.1 the x-axis displays the departing moments from Lisbon for the destination Berlin, where the y-axis shows the trip's total cost (including the returning trip) and each colored bar represents a different driver. Analyzing the first departing moment (2017-01-15 00:00) there are 3 clusters of drivers with the same costs. That is, for the same departing moment there are multiple drivers that perform the trip with the same total cost, then in turn are grouped in 3 large clusters. This clusters are an indication that there is a chance that these drivers, that share equal costs and equal departing moments can possibly be considered equals. The second departing moment (2017-01-16 00:00) as all 4 drivers performing the trip with the same cost. And finally the last departing moment (2017-01-17 00:00) is similar to the first

[1]Due to its size, Table 5.1 is available as an annex.

departing moment, despite being fewer drivers, and at lower costs, they still form groups of equal costs. There are a total of 9 unique combinations of total cost and departing moments in this figure, which means that taking into consideration this figure alone, the 51 drivers could be grouped into 9 unique clusters. From which can be extracted 9 model drivers, or, types of driver.

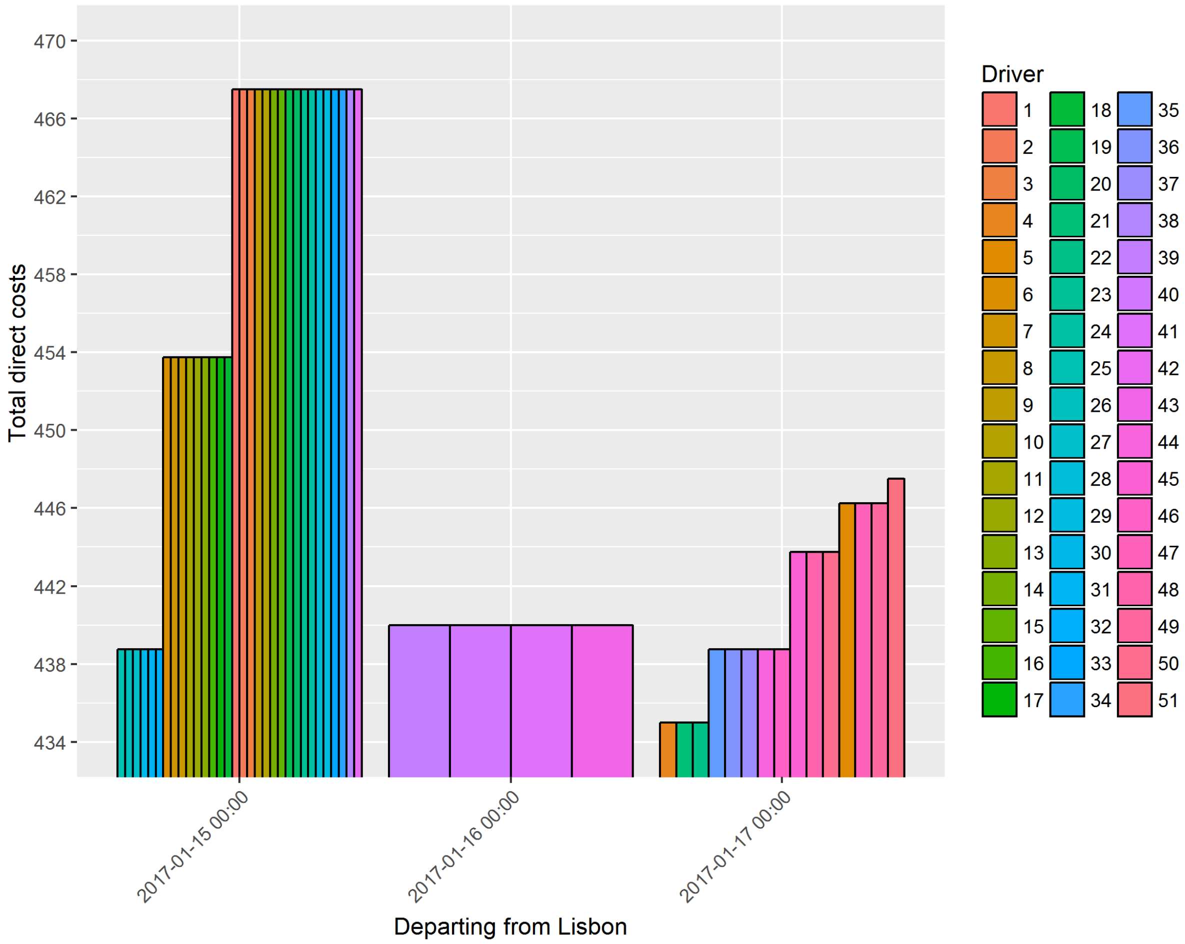

Figure 5.1: Plot of total direct costs by departing (Lisbon), of 51 drivers in 1st schedule, to Berlin.

The y-axis in Figure 5.2 is the same as in Figure 5.1, while the x-axis displays the arriving moments to the destination Berlin, and the bars continue to represent the different drivers. The number of x-axis values increased to 8, meaning that many of the drivers in a cluster on the previous figure are no longer grouped, they differed. And so, the number of unique combinations between arriving moments and total cost are here 12, which implies 12 types of drivers. No notice that in the arriving moment 2017-01-23 01:00 one of the drivers has a total trip cost of around 446, two other drivers have the same total cost but with a different arriving moment (2017-01-32 05:00) Since the arriving moments are different, these are considered two different unique combinations. In a scenario were only departing would matter, the number of driver's type found in Figure 5.1 would suffice. But since arriving to destination is also an important criterion it is taken into consideration the larger number of types of drivers among all situations. The farther the trip the more likely is that drivers differ.

Figure 5.3 is very similar to Figures 5.1 and 5.2, but in this case the x-axis represents the returning moment. The number of moments have increased to 10, and with exception to the returning moment

2017-01-30 01:00, there is only 1 total cost per arriving moment. In these case there are now 11 unique combinations of returning moments and total cost. Since in Figure 5.2 there were 12 different combinations, this means that some driver/drivers have regrouped again. In the 3 key moments (departing, arriving and returning) there were a maximum of 12 unique combinations and that is the number of drivers type that would be consider since using 11 or 9 would imply losing important information. That is, if only having into consideration these plots, and ignoring possible changes that might occur between plots.

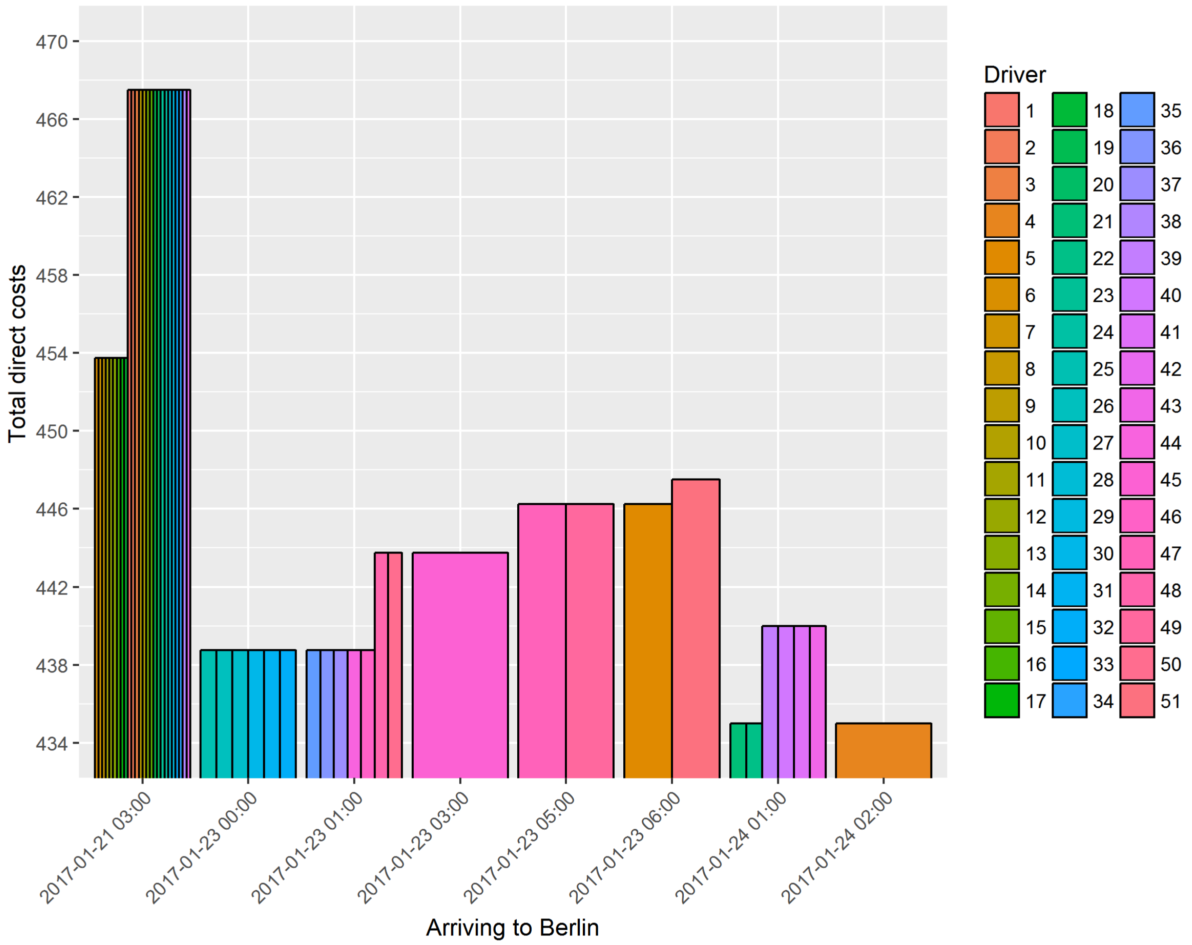

Figure 5.2: Plot of total direct costs by arriving (Berlin), of 51 drivers in 1st schedule, to Berlin.

In this description it was only used the plots as a reference, so it is arguable that even if the total number of unique combinations was equal in all 3 plots that did not mean that that number would be the correct one, since drivers could swap clusters between plots. For example assuming that in the first plot there were 2 clusters, cluster A and cluster B, and on the second plot there were also 2 clusters, cluster C and cluster D. This could mean that all drivers on cluster A would be on cluster C, and that those on cluster B would all be on cluster D, having only 2 types of drivers, AC and BD. But it could also mean that some drivers of cluster A would be on cluster C and some on cluster D, and the same for cluster B. Having clusters A and B swap some of their drivers. Being possible the combinations AC, AD, BC and BD. It would still be possible to read this information by analyzing each driver bar and it behavior from plot to plot. But 51 drivers in 51 different (but enough) colors would make that a hard task. The purpose here was to present the reasoning being the typification of drivers and for the reader to have a more direct perception by use of plots. As explained on chapter 4, the program was built considering a ”unique combination” as a combination of total cost + departing moment + arriving moment + returning moment which is more exact and prevents the situation described. The result was coincidentally 12 type of driver.

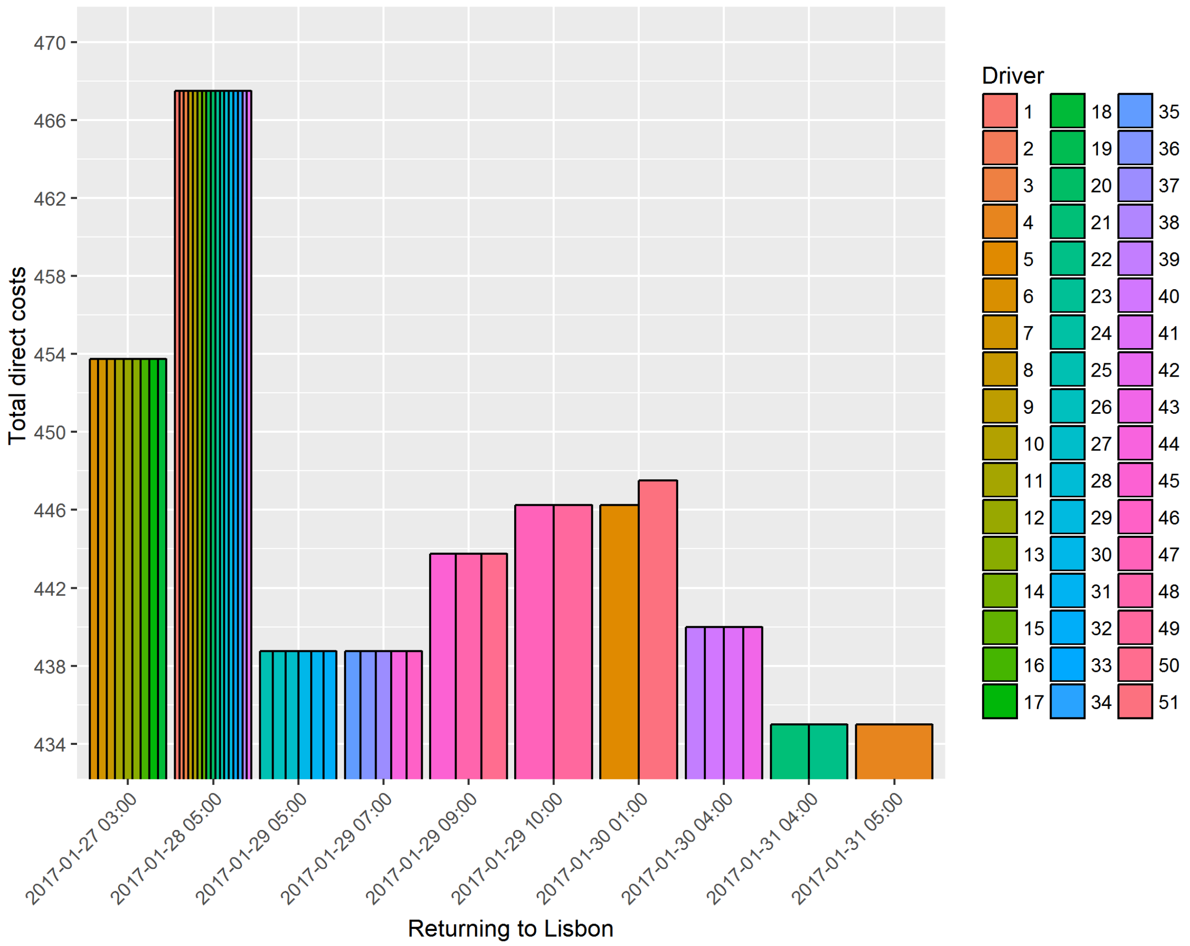

Figure 5.3: Plot of total direct costs by returning (back to Lisbon), of 51 drivers in 1st schedule, to Berlin.

Now that the number of unique drivers has been found, Table 5.2 was created, where per each unique driver (column 3) it is displayed who is the driver that represents such type (1^{st} column) and how many similar drivers are available (2^{nd} column). The later information will be required for the optimization phase.

Table 5.2: The outputted types of drivers' table.

Driver	Number of drivers	Driver type
1	17	1
4	1	2
5	1	3
6	9	4
21	2	5
25	6	6
35	5	7
39	4	8
45	1	9
47	2	10
48	2	11
51	1	12

5.2 Singular drivers

After establishing the type of drivers, it was time to calculate all the information for drivers driving alone, i.e.: singular drivers. For that, the drivers history is required, but since the drivers have been typified, the 51 drivers history was no longer needed. However a shorter version (Table 5.3) with the history of the drivers that represent the types of drivers was produced, and used starting from this point.

The first conclusion to be taken at a global glance over all plots[2], is that it is frequently possible to have different available departing or arriving periods for the same cost. I.e.: a particular cost can have multiple options of departure. In other words, it is possible for a driver to depart at different moments without that having any implication on the total cost, and equivalent for the arriving moment, including both arriving to the destination and arriving back from the trip. The same is also true for a constant moment, it is possible to have multiple different costs.

For all 3 destinations, drivers type 1 and 4 are the fastest to leave (Figures 5.4, 5.7, and 5.10), but also the most expensive. Driver type 6 is also one of the fastest to leave, but only on one occurrence which is suspicious. It is likely that it is a driver that is able to depart immediately but has to stop very soon for being in a situation were it reaches a constrain quickly. Looking at the tables that have the information displayed in these plots, (that are too big to be part of this text), driver type 6 can work for 1 hour if it departs at the 00:00 hours of 2017-01-15 but it has stop immediately after, being able to reach the first nearest frontier city only two days later, this happens because the driver has to stop in order to respect the weekly rest of 45 continuous hours. The pattern of departures is relatively constant as it is expected since the availability to depart is independent of the destination.

[2]Due to their sizes, the plots are available as annexes.

Table 5.3: The first 32 rows of the drivers history filtered by types of drivers.

	1	2	3	4	5	6	7	8	9	10	11	12
2017-01-01 00:00	0	0	0	1	1	1	0	0	0	0	0	1
2017-01-01 01:00	0	0	0	1	0	0	0	1	0	0	0	1
2017-01-01 02:00	0	0	0	1	0	0	0	1	0	0	0	0
2017-01-01 03:00	0	0	0	1	0	0	0	1	0	0	0	0
2017-01-01 04:00	0	0	0	1	0	0	0	1	0	0	0	0
2017-01-01 05:00	0	0	0	0	0	0	0	1	0	1	0	0
2017-01-01 06:00	0	0	0	1	0	0	0	0	1	1	0	0
2017-01-01 07:00	0	0	0	1	0	0	0	1	1	1	0	0
2017-01-01 08:00	0	0	0	1	0	0	0	1	1	1	0	0
2017-01-01 09:00	0	0	0	1	0	0	0	1	1	0	1	0
2017-01-01 10:00	0	0	0	0	0	0	0	1	0	0	1	0
2017-01-01 11:00	0	0	0	0	0	0	0	0	0	0	1	0
2017-01-01 12:00	0	0	0	0	0	0	0	0	0	0	1	0
2017-01-01 13:00	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 14:00	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 15:00	0	0	0	0	0	0	1	0	0	0	0	1
2017-01-01 16:00	0	0	0	0	0	0	1	0	0	0	0	1
2017-01-01 17:00	0	0	0	0	0	0	1	0	0	0	1	1
2017-01-01 18:00	0	0	0	0	0	0	1	0	0	0	1	1
2017-01-01 19:00	0	0	0	0	0	0	0	0	0	0	1	0
2017-01-01 20:00	0	0	0	0	0	0	1	0	0	0	1	0
2017-01-01 21:00	0	0	0	0	0	0	1	0	0	0	1	0
2017-01-01 22:00	0	0	0	0	0	0	1	0	1	1	0	0
2017-01-01 23:00	0	0	0	0	0	0	1	0	1	1	1	0
2017-01-02 00:00	0	0	0	1	0	0	1	1	1	1	1	0
2017-01-02 01:00	0	0	0	1	0	0	0	1	1	1	0	0
2017-01-02 02:00	0	0	0	1	0	0	0	1	0	0	0	1
2017-01-02 03:00	0	0	0	1	0	0	0	1	0	0	0	1
2017-01-02 04:00	0	0	0	1	0	0	0	1	0	0	0	1
2017-01-02 05:00	0	0	0	0	0	0	0	0	0	0	0	1
2017-01-02 06:00	0	0	0	1	0	0	0	1	0	0	0	0
2017-01-02 07:00	0	0	0	1	0	0	0	1	1	1	0	0

Driver type 8 is one of the fastest to be able to depart but with a decrease in costs when compared with the drivers type 1 and 2. Unfortunately it is the latest or among the latest to arrive to the destination (Figures 5.5, 5.8 and 5.11), and the same when returning from the trip (Figures 5.6, 5.9 and 5.12).

The remaining types of drivers depart later at lower costs and are very diverse on their arriving moments, for both destination and returning.

As previously conducted, this calculation was also fulfilled in a dedicated Linux machine and lasted for 1.981148 hours.

5.3 Co-op drivers

Calculating the co-op drivers schedules is not a procedure that needs to occur after the singular drivers calculation. Since the inputs required are equal, both can be performed in simultaneously. Meaning an increase in global performance, although it would be needed a second computer to do so.

As for the singular drivers, the drivers' history used for co-op drivers is the same as the one presented in Table 5.3.

Due to large amounts of data, the plots will not be analyzed since the display of these plots required large pages for the data to be identifiable. Furthermore, the display of tables is not an option either, for each table has more than 3,000 lines. The idea is the same as presented with the singular drivers, the figures presented are merely an illustration of the data, even the manager or decision maker will have very little use for these plots. Producing the tables is, for security reasons, a good idea, in the case of an interruption of the program it is always possible to resume the calculations as long as the tables have been produced. Since the program requires some time to perform all computations accidents may occur, like power failures that can turn the computer off.

This process took 1.27799 days[3] to be calculated on a dedicated Linux machine.

Table 5.4 displays the time it took for the program to calculate each of the different schedules calculation's phases.

Table 5.4: Processing times.

Types of driver	5.700763	mins
Singular drivers	1.981148	hours
Co-op drivers	1.27799	days

5.4 Complete resume matrices

After both Singular and Co-op drivers have been calculated, the information was then complied in the two complete resume matrices for costs and for time of arrival, just like Tables 4.3 and 4.4 presented on chapter 4, but due to their size are not presented here. The resume matrix for time was then adjusted for each destination by converting the soonest arriving date to a 0 cost, arriving 1 hour later than that to 1 unit of cost, arriving 2 hours later than the best possible time, to 2 units of cost, and so on[4].

[3] It is not a typo, it took more than a complete day to calculate 10,368 different schedules.

[4] The information displayed on Tables 4.3, 4.4 and 4.5 is actually the information mentioned here.

5.5 Optimization

It was determined that the demand for the first destination was of 5 shipments, and additionally, the second and third destinations were both of 3 shipments each. The weight given to costs and time was of 50% each, making it an alpha of 0.5.

The best found solution (Table 5.5) was to allocate 5 drivers of type 1 to Madrid, in schedule 1. By leaving at midnight of the 15^{th}, arriving to its destination 10 hours later, and returning in the next day at 08:00 with indirect costs of 1 unit, which means that it arrives to destination only 1 hour later than what is possible. This results in a direct cost of 110 units making it a 111 units of global costs per driver sent to this destination. Recalling the plots from destination Madrid[5] allocating drivers type 1 is within expectations, since this type of driver is faster despite the costs being higher than the other drivers.

To Paris, 2 sets of drivers type 7 and 8 are allocated in a co-op method, where driver type 7 is the starting driver, in schedule 24, producing a global cost of 263.5 units per set of drivers type 7 and 8. Moreover, another set of co-op drivers is also sent to destination Paris, this being, 2 drivers type 8, in schedule 24, costing a global 263.25 per set.

Finally, for Berlin are sent 3 sets of 2 drivers type 1 in co-op method, in schedule 24. As a result the global cost for this destination is of 408.75 per set sent. Therefore, the total indirect costs amount to 124 units, and the total direct costs to 2447.5 units, globally amounting to 2571.5 units of expenses to satisfy the demand of all destinations.

Table 5.5: A resume table of the optimization solution and derived values.

Driver	Schedule	Madrid	Paris	Berlin	Departing	Arriving	Returning	Indirect costs	Direct costs
1	1	5			2017-01-15 00:00	2017-01-15 09:00	2017-01-16 08:00	1.00	110.00
7.8	24		2		2017-01-16 00:00	2017-01-17 09:00	2017-01-19 15:00	26.00	237.50
8.8	24		1		2017-01-16 00:00	2017-01-17 05:00	2017-01-19 20:00	22.00	241.25
1.1	24			3	2017-01-15 23:00	2017-01-17 18:00	2017-01-19 20:00	15.00	393.15

Table 5.6 shows the schedule that driver type 1 will take when driving to Madrid. Table 5.7 shows the selected schedules that two drivers types 1 will take when driving to Berlin. The first driver type 1 starts to drive at 23:00 hours of the 15^{th} and continues until the 04:00 hours of the following day, when the second driver type 1 will then start driving. While the second driver type 1 is driving, the first driver is resting, and when the second driver needs to rest the first driver (already rested) returns to drive (since no other restriction forbids it), which occurs at the 09:00 hours of the 16^{th}. On the 18^{th} the first driver cannot start to drive at 00:00 hours as it did on the previous day, because it hasn't had its 11 continuous hours of rest, only at 01:00 hours it has finished this rest period and is then allowed to drive. For similar reasons the same occurs on the following day, then when both drivers return home at the 20:00 hours.

After sending these drivers, the amount of available drivers changes (Table 5.2). The updated table of available drivers and their types is as presented on Table 5.8.

As it is showed by Table 5.8 there were used 11 drivers of type 1, remaining 6 available, probably due to the quickest availability of this type of driver. Relatively to driver type 7, there were used 4 of a total of

[5] Figures 5.4, 5.5 and 5.6.

Table 5.6: Selected schedule 1 of driver type 1 for destination 1 (Madrid).

	2017-01-15	2017-01-16
00:00	1	1
00:00	1	1
01:00	1	1
02:00	1	1
03:00	1	1
04:00	0	0
05:00	1	1
06:00	1	1
07:00	1	1
08:00	1	0
09:00	0	0
10:00	0	0
11:00	0	0
12:00	0	0
13:00	0	0
14:00	0	0
15:00	0	0
16:00	0	0
17:00	0	0
18:00	0	0
19:00	0	0
20:00	0	0
21:00	0	0
22:00	0	0
23:00	0	0

5 available drivers. And finally, drivers of type 8 were all used. This, however, is not unexpected as seen by the plots, driver type 8 is a very interesting option, since it departs relatively fast and at acceptable costs, although it is not the cheapest nor the fastest, it is the one who returns the best overall solution given the criteria. It is only speculative if more drivers of type 8 would be selected if available.

Starting over on a following day requires that all the process to be recalculated from the beginning, including studying the typology of drivers, meaning that the updated types of driver table might lose its usability, since another day of idle work for the drivers means that constraints are now less likely to be reached. Of course that of the 51 original drivers, only 34 remained, but more drivers can (and will) return throughout time. Even if no drivers are sent, the adding drivers and rest period will affect the typology and consequent calculations. So each day, assuming that 24 schedules are being calculated, the whole heuristic must be recalculated.

Table 5.7: Selected schedule 24 of two (different) drivers type 1 for destination 3 (Berlin).

	2017-01-15		2017-01-16		2017-01-17		2017-01-18		2017-01-19	
	1	1	1	1	1	1	1	1	1	1
00:00	0	0	1	0	1	0	0	0	0	0
01:00	0	0	1	0	1	0	1	0	0	0
02:00	0	0	1	0	1	0	1	0	1	0
03:00	0	0	1	0	1	0	1	0	1	0
04:00	0	0	0	1	1	0	1	0	1	0
05:00	0	0	0	1	0	1	1	0	1	0
06:00	0	0	0	1	0	1	0	1	1	0
07:00	0	0	0	1	0	1	0	1	0	1
08:00	0	0	0	1	0	1	0	1	0	1
09:00	0	0	1	0	0	1	0	1	0	1
10:00	0	0	1	0	1	0	0	1	0	1
11:00	0	0	1	0	1	0	1	0	0	1
12:00	0	0	1	0	1	0	1	0	1	0
13:00	0	0	0	1	1	0	1	0	1	0
14:00	0	0	0	1	0	1	1	0	1	0
15:00	0	0	0	1	0	1	0	1	1	0
16:00	0	0	0	1	0	1	0	1	0	1
17:00	0	0	0	0	0	1	0	1	0	1
18:00	0	0	0	0	0	0	0	1	0	1
19:00	0	0	0	0	0	0	0	0	0	1
20:00	0	0	0	0	0	0	0	0	0	0
21:00	0	0	0	0	0	0	0	0	0	0
22:00	0	0	0	0	0	0	0	0	0	0
23:00	1	0	0	0	0	0	0	0	0	0

Table 5.8: The types of driver's table, updated after the optimization.

Driver	Number of drivers	Driver type
1	6	1
4	1	2
5	1	3
6	9	4
21	2	5
25	6	6
35	3	7
39	0	8
45	1	9
47	2	10
48	2	11
51	1	12

5.6 Costs' Weighting

In the results obtained previously, the alpha used was of 0.5, giving equal weight for both direct costs, and indirect costs (arriving time related costs). Depending on how the company defines this weight distribution, the solution may vary.

When $\alpha = 0$, the solution is focused only on the time costs, making the direct costs higher than what they could be. On the other hand, when $\alpha = 1$, the solution optimization focus only on the direct costs, making the drivers arriving later than what they could.

On the rows of Table 5.9, are presented the different solutions for different values of alpha, where the first column is the interval of alpha. The second and third columns represent the direct and indirect costs for that solution, and the last column is the sum of all costs.

For this particular case, alpha can be grouped as showed in the first column. There is no difference on the solutions when alpha remains within an interval, meaning that the alpha is not a continuous variable. E.g.: when alpha is between 0.001 and below 0.09, the solution is the same, that is, it focus mostly on the indirect costs. Although the solution for this interval maintains a 0 cost for indirect costs, the direct costs are different from the previous interval, [0:0.001[, because the solutions are indeed different, it just so occurs that the indirect costs could remain unchanged.

Looking at the total costs column, the minimum value is 2571.50, this occurs for two different solutions, when alpha is on the intervals [0.470:0.501[and [0.501:0.505[. Both intervals have a relative balance between direct and indirect costs, implying that in this case, assigning similar weight to both costs will return the minimum solution.

Figure 5.13[6] is a plot of the information on Table 5.9, where the alpha intervals seem to form a certain curvature.

[6]Due to its size, Figure 5.13 is available as an annex.

Table 5.9: Costs grouped by alpha interval.

Alpha intervals	Direct costs	Indirect costs	Total costs
[0:0.001[	2708.75	0.00	2708.75
[0.001:0.091[	2690.00	0.00	2690.00
[0.091:0.118[	2650.00	4.00	2654.00
[0.118:0.138[	2642.50	5.00	2647.50
[0.138:0.286[	2623.75	8.00	2631.75
[0.286:0.325[	2601.25	17.00	2618.25
[0.325:0.359[	2588.75	23.00	2611.75
[0.359:0.371[	2538.75	51.00	2589.75
[0.371:0.460[	2515.00	65.00	2580.00
[0.460:0.470[	2495.00	82.00	2577.00
[0.470:0.501[	2447.50	124.00	2571.50
[0.501:0.505[	2417.50	154.00	2571.50
[0.505:0.517[	2335.00	238.00	2573.00
[0.517:0.521[	2331.25	242.00	2573.25
[0.521:0.608[	2313.75	261.00	2574.75
[0.608:0.616[	2295.00	290.00	2585.00
[0.616:0.624[	2293.75	292.00	2585.75
[0.624:0.667[	2275.00	323.00	2598.00
[0.667:0.733[	2260.00	353.00	2613.00
[0.733:0.791[	2251.25	377.00	2628.25
[0.791:0.828[	2242.50	410.00	2652.50
[0.828:1]	2240.00	422.00	2662.00

6: Conclusions and further research

The main objective was achieved, the heuristic calculates several schedules for several drivers which are then used to reach a global optimal solution given the demand. The program is effective but not efficient. It has its flaws and drawbacks, there are many aspects that can be approached differently from how they were in this project. These aspects include, among others, what to optimize or when, or how the costs are calculated. There is also a lot of room for improvement but there is already great adaptability. Overall, it is a decent proof of concept with some usability.

The next section is an analysis of the program's efficiency. Followed by a section highlighting the importance of the costs in the project. And the last section will be dedicated to some suggestions regarding future researches.

6.1 Efficiency

It is necessary more than a full day for the process to be complete, even when using a relatively small amount of typified drivers. This may be due to the use of R software, or even just due to poor code. It was studied the possibility of reproducing the code in another language, one such language was Python. Some tests were conducted but the improvement in speed didn't seem to be significant. Another possibility was to use C++, a lower level language, hard to work with but with good possibility of improvement. Unfortunately, C++ is a more complex language when compared to Python, which implies requiring even more time for development. Moreover, Python somewhat similar to R, would still need time to develop and in the end the improvement could not be enough.

It is also possible to improve the program efficiency by way of hardware, this is not desirable but it is a possibility. Given the design of the program, it is always possible to increase the number of hardware processors in order to increase speed. A possible solution for this would be mini computers, like the "raspberry pi", since these are cheap and the program does not require a powerful processor to work, but the use of multiple processors would decrease drastically the time for calculations. An alternative solution would be to use Ad Hoc service.

For time limitations it was decided to publish the inefficient work, rather than risk not publishing anything at all.

A more direct approach would be to use integer programming to solve the whole process. But, as inefficient as the heuristic may be, using integer programming on a dedicated (and not open sourced) software would most likely be even more inefficient, particularly with large amounts of data.

For the purpose of the example, co-op drivers were used that can also be available as single drivers, and reciprocally, which took a long time. A company may not want to study every type of driver as both options, but have a set of fixed drivers that only drive alone and another fixed set of drivers that only work in a co-op system, dramatically decreasing the calculation time of the whole process.

Accordingly, when efficiency is not a problem, more constraints can be added, and the intervals of time can be narrower. Regardless of its inefficiency, this project is quiet robust, being able to adapt to many changes that may be desired. It is still useful, especially for a low number of drivers.

Additionally typifying drivers is a great cut on useless calculations, since calculating for 12 drivers took more than a day, calculating for 51 drivers would surely take more than a week. To note that in the example given in this project the typification of drivers had 4 types of drives (drivers type 2, 3, 9 and 12)[1] with only 1 real driver available. This means that calculating co-op driving schedules for doubles[2] in these types of drivers was unnecessary since a driver type with 1 singular element can't drive with itself (for lack of more elements). In this case it means that 288 calculations occurred in vain and could be avoid, making the program leaner. It does not have a huge negative impact but it is something that can be improved. How the typification is fulfilled is a study worth for itself. Unless two drivers have the exact same history, typifying drivers will always incur in a risk of ignoring information that may be important. How much this ignored information is, depends on how far the destinations are, and how many constraints there are. If, on the other hand, a faster method that works with all information, e.g.: brute force, is available, then typifying loses its importance. Otherwise, the impact of the total number of different drivers and the destinations' distance, on the typification, is an interesting topic. Assuming that a brute force approach is not possible by code improvement or using another language, then this is a promising subject to be studied. A very interesting approach to be taken is the use of machine learning.

6.2 Costs

The calculation of costs per driver may be done in a different way. Each company must have a form of payment calculated based on the hours the driver has driven, but also where the driver drives. Consequently the method used in this project may not make the most sense for most companies if for any. It was just a representation that can be adapted.

The weight (alpha) of what costs matter the most for the company, is also an interesting theme to be studied. What costs should have a bigger weight? Should this weight be adjustable along the year? And if so, when? These are just a few interesting questions that can be answered by exploring this topic.

In the example, it was showed a mix of direct costs, with costs relative to arriving moments. Instead, it could be a different mix, e.g.: direct costs plus arriving to destination costs plus returning home costs. The weight given to each cost can be adjusted too. Moreover, how the indirect costs are calculated can also be changed. Perhaps it would make more sense to define a standard time per destination dependable on the day the order was placed, therefore, the costs would be a reflection of how late the shipment arrives compared to this standard date. Instead of having linear growing costs, maybe it would be more realistic to have exponential growing costs. Another way to calculate costs would be to aggregate several moments, e.g.: arriving between the 00:01 and the 12:00 has a cost, and arriving between the 12:01 and the 24:00 has a different cost. With the ability to adjust these intervals as necessary. This would certainly be a very interesting subject to dive in.

[1] Table 5.2.

[2] A pair of two equal driver types, e.g.: driver type 1 with driver type 1; driver type 4 with driver type 4.

The heuristic was built with a mindset on driving as soon as possible regardless of what costs that would entail. An interesting approach would be to develop an alternative heuristic that initially would focus on minimizing costs, and in a second version, an heuristic that would combine a mix of driving as soon as possible and at minimum cost.

6.3 Suggestions

Being able to calculate different schedules and optimizing the allocation is only half the solution, it is then necessary to find the best long term strategy. If the process was efficient, the second phase of this project would have been to run several simulations of different strategies, and then comparing the results. Therefore, the first simulation would be sending an available driver as soon as an order is placed, with as few calculations as possible. Perhaps a priority would be to send the most rested driver, but nothing more. This would be the standard strategy. A second strategy would be to wait a *t* period of time, make the calculations (which would be done quite fast), and then allocate the drivers to the different destinations. A third strategy would be to wait a bigger *t* before calculating the allocation. And finally, a last strategy would be to wait *t* equal to the second strategy, and allocate a driver, but if that driver hasn't been sent yet, allow the next calculation to consider this driver "eligible" for allocation. Comparing different strategies by simulation is a good form of strategic analysis, for it gives a decent view of each strategy to the management. Different strategies will likely have different performances, a particular strategy can seem too expensive in a short term, but in a long term be the best option. It is also possible that a new strategy may be put together by merging multiple strategies. For example a strategy 1 and strategy 2, where strategy 1 performs better on winter, but then strategy 2 performs better on the summer. A 3rd strategy could then be a mix of strategy 1 for winter season and then strategy 2 on summer, gaining a better performance in the long term.

A different, but very interesting alternative method of work, would be the use of drivers with fixed routes in fixed regions, where the cargo would work as a baton[3], passed on among the drivers. This way the cargo could be always on the move since a driver would pass it on to another available driver when it would be about to reach the limit of its constraints. Adding to this, there would be cargo traveling both ways in a route. A driver could take cargo from a destination A to a point B, pass the cargo to a second driver that would take this cargo to a point C, and in the mean time, the first driver would take the second driver's cargo to destination A. Exploring this method by ways of simulation would be of great interest, especially for companies with large amount of drivers and shipments traveling in a equilibrium way for both ends of a route.

Another interesting subject to be studied would be the goal to minimize idle time for trucks. A truck is a great investment for a company, and every time a truck is motionless, it results on an investment that is not paying off. While self-driving trucks are not available yet, using multiple drivers to have the trucks moving at a maximum of time would be an appealing research. Especially if a driver is paid based on driven hours just as in this project.

Although it was ignored, cabotage is a reality for many companies, especially big companies. If such companies work in a form of clusters the program is usable as it is, but when involving drivers that may not be allocated to a cluster and are able to operate, even if in a casual moment, for a different country

[3]A baton is the object passed on by the athletes on a relay race.

that they are passing trough or near, hence the term cabotage, the program could be more useful if it would have the capacity to deal with this data.

Despite the fact that the focus of the project was freight transportation, the heuristic can be adapted to people transportation and even possibly to other areas that have constraints on the allocation of workers.

Bibliography

[1] Frada, João, **Novo Guia Prático** para Pesquisa, Elaboração e Apresentação de Trabalhos Científicos e Organização de Currículos, 1.^{a} edição - 13.^{a} reimpressão, Edições CLINFONTUR CLIN. MED, L^{DA}, 1991 - 2008, 190 p.

[2] ≪Holidays in Portugal in 2017≫,
http://www.timeanddate.com/holidays/portugal/
Consulted on 2017-01-12.

[3] European Commission, Mobility and Transport. ≪Driving time and rest periods≫,
http://ec.europa.eu/transport/modes/road/social_provisions/driving_time/index_en.htm
Consulted on 2016-04-19.

[4] EUR-Lex Access to European Union law. ≪REGULATION (EC) No 561/2006 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 15 March 2006≫,
http://eur-lex.europa.eu/eli/reg/2006/561/2015-03-02
Consulted on 2016-04-19.

[5] ≪Subsídio de Turno e Trabalho Noturno≫,
https://www.economias.pt/subsidio-de-turno-e-trabalho-noturno/
Consulted on 2016-10-06.

[6] Comissão para a igualdade no trabalho e no emprego. ≪Descanso diário≫, N.^{o}1 do Artigo 214.^{o} do Código do Trabalho, Lei N.^{o}7/2009.
http://www.cite.gov.pt/pt/legis/CodTrab_L1_005.html
Consulted on 2016-11-03.

[7] Direitos do Trabalho. ≪Compensação por trabalho extraordinário≫,
http://meusalario.pt/inicio/direitos-do-trabalho/compensacao
Consulted on 2016-10-06.

[8] Data elements and interchange formats. ≪ISO 8601≫,
https://en.wikipedia.org/wiki/ISO_8601
Consulted on 2016-11-28.

[9] Traffic bans. ≪Holidays≫,
http://trafficban.com/country.portugal.holidays.153.en.html
http://trafficban.com/country.spain.holidays.75.en.html
http://trafficban.com/country.france.holidays.56.en.html
http://trafficban.com/country.germany.holidays.135.en.html
Consulted on 2017-01-02.

[10] Traffic bans. ≪Local Constraints≫,
http://trafficban.com/country.portugal.info.153.en.html
http://trafficban.com/country.france.info.56.en.html
http://trafficban.com/country.germany.info.135.en.html
Consulted on 2016-10-22.

[11] Driving constraints in Europe on specific days and times. ≪Local Constraints≫,
http://www.transsib-logistics.de/en/drivingbans
Consulted on 2016-10-21.

[12] eurostat. ≪Annual road freight transport vehicle movements, loaded and empty, by reporting country (Mio Veh-km, 1 000 Jrnys)≫,
http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=road_go_ta_vm&lang=en
Consulted on 2017-06-15.

[13] Motor Transport Institute. ≪The Impact of Regulation of the Road Transport Sector on Entrepreneurship and Economic Growth in the European Union≫,
https://mib.gov.pl/files/0/1797568/REPORTITSEN2017.pdf
Consulted on 2017-06-15.

[14] European Commission, Mobility and Transport. ≪An Overview of the EU Road Transport Market in 2015≫,
https://ec.europa.eu/transport/sites/transport/files/mobility-package-overview-of-the-eu-road-transport-market-in-2015.pdf
Consulted on 2017-06-15.

[15] European Commission, Mobility and Transport. ≪Statistical pocketbook 2016≫,
https://ec.europa.eu/transport/sites/transport/files/pocketbook2016.pdf
Consulted on 2017-06-15.

[16] Eurostat statistics-explained. ≪Glossary:Tonne-kilometre (tkm)≫,
http://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Tonne-kilometre_(tkm)
Consulted on 2017-06-19.

Appendix

Table 4.2: Example of the Resume Matrix from Lisbon to Madrid.

Driver	Schedule	Lisbon	Badajoz	Madrid	Returning	Total time to destination	Total time on trip	Total direct cost
1	1	2017-01-15 00:00	2017-01-15 03:00	2017-01-15 09:00	2017-01-16 08:00	9.00 hours	1.33 days	110.00
1	2	2017-01-15 01:00	2017-01-15 04:00	2017-01-15 10:00	2017-01-16 08:00	9.00 hours	1.29 days	110.00
1	3	2017-01-15 02:00	2017-01-15 05:00	2017-01-15 11:00	2017-01-16 08:00	9.00 hours	1.25 days	110.00
1	4	2017-01-15 03:00	2017-01-15 06:00	2017-01-15 12:00	2017-01-16 08:00	9.00 hours	1.21 days	110.00
1	5	2017-01-15 04:00	2017-01-15 07:00	2017-01-15 13:00	2017-01-17 08:00	9.00 hours	2.17 days	110.00
1	6	2017-01-15 05:00	2017-01-15 08:00	2017-01-15 14:00	2017-01-17 08:00	9.00 hours	2.12 days	110.00
1	7	2017-01-15 06:00	2017-01-15 09:00	2017-01-15 15:00	2017-01-17 08:00	9.00 hours	2.08 days	110.00
1	8	2017-01-15 07:00	2017-01-15 10:00	2017-01-15 16:00	2017-01-17 08:00	9.00 hours	2.04 days	110.00
1	9	2017-01-15 08:00	2017-01-15 11:00	2017-01-15 17:00	2017-01-17 08:00	9.00 hours	2.00 days	110.00
1	10	2017-01-15 09:00	2017-01-15 12:00	2017-01-15 18:00	2017-01-17 08:00	9.00 hours	1.96 days	110.00
1	11	2017-01-15 10:00	2017-01-15 13:00	2017-01-15 19:00	2017-01-17 08:00	9.00 hours	1.92 days	110.00
1	12	2017-01-15 11:00	2017-01-15 14:00	2017-01-15 20:00	2017-01-17 08:00	9.00 hours	1.88 days	110.00
1	13	2017-01-15 12:00	2017-01-15 15:00	2017-01-15 21:00	2017-01-17 08:00	9.00 hours	1.83 days	110.00
1	14	2017-01-15 13:00	2017-01-15 16:00	2017-01-15 22:00	2017-01-17 08:00	9.00 hours	1.79 days	110.00
1	15	2017-01-15 14:00	2017-01-15 17:00	2017-01-15 23:00	2017-01-17 07:00	9.00 hours	1.71 days	111.25
1	16	2017-01-15 15:00	2017-01-15 18:00	2017-01-16 00:00	2017-01-17 07:00	9.00 hours	1.67 days	110.00
1	17	2017-01-15 16:00	2017-01-15 19:00	2017-01-16 01:00	2017-01-17 07:00	9.00 hours	1.62 days	108.75
1	18	2017-01-15 17:00	2017-01-15 20:00	2017-01-16 02:00	2017-01-17 07:00	9.00 hours	1.58 days	107.50
1	19	2017-01-15 18:00	2017-01-15 21:00	2017-01-16 03:00	2017-01-17 07:00	9.00 hours	1.54 days	106.25
1	20	2017-01-15 19:00	2017-01-15 22:00	2017-01-16 04:00	2017-01-17 07:00	9.00 hours	1.50 days	106.25
1	21	2017-01-15 20:00	2017-01-15 23:00	2017-01-16 05:00	2017-01-17 07:00	9.00 hours	1.46 days	105.00
1	22	2017-01-15 21:00	2017-01-16 00:00	2017-01-16 06:00	2017-01-17 07:00	9.00 hours	1.42 days	102.50
1	23	2017-01-15 22:00	2017-01-16 01:00	2017-01-16 07:00	2017-01-17 07:00	9.00 hours	1.38 days	100.00
1	24	2017-01-15 23:00	2017-01-16 02:00	2017-01-16 08:00	2017-01-17 07:00	9.00 hours	1.33 days	97.50
2	1	2017-01-17 00:00	2017-01-17 03:00	2017-01-17 09:00	2017-01-18 08:00	9.00 hours	1.33 days	95.00
2	2	2017-01-17 01:00	2017-01-17 04:00	2017-01-17 10:00	2017-01-18 08:00	9.00 hours	1.29 days	93.75
2	3	2017-01-17 02:00	2017-01-17 05:00	2017-01-17 11:00	2017-01-18 08:00	9.00 hours	1.25 days	93.75

Table 5.1: The 1st day of the first 31 out of 51 drives' history.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2017-01-01 00:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	1	1	0	0	1	1
2017-01-01 01:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
2017-01-01 02:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
2017-01-01 03:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
2017-01-01 04:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
2017-01-01 05:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
2017-01-01 06:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 07:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
2017-01-01 08:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
2017-01-01 09:00	0	1	1	0	0	1	1	1	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
2017-01-01 10:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
2017-01-01 11:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 12:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 13:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 14:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 15:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2017-01-01 16:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2017-01-01 17:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2017-01-01 18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2017-01-01 19:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017-01-01 20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2017-01-01 21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2017-01-01 22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2017-01-01 23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

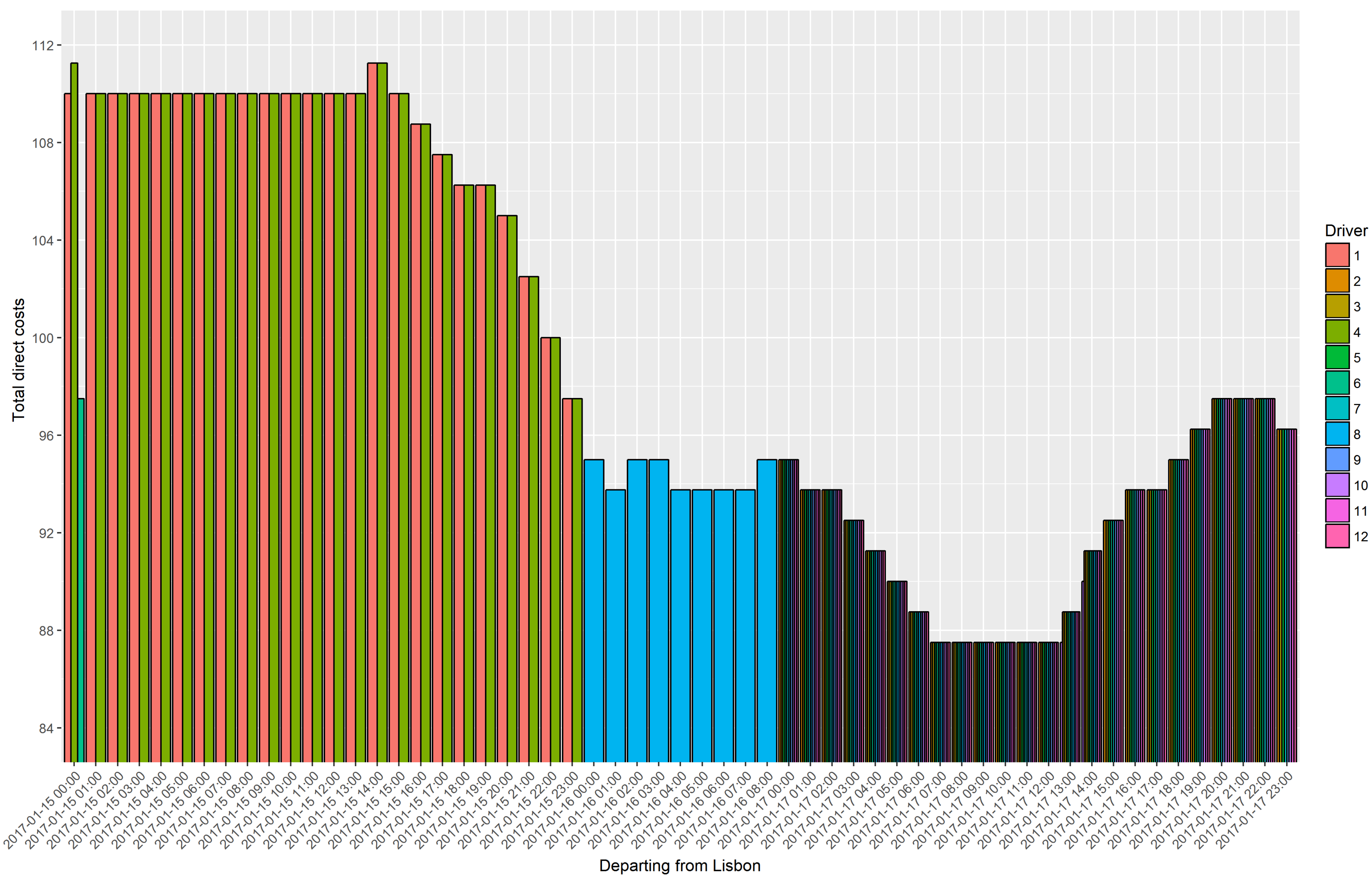

Figure 5.4: Plot of total direct costs by departing (Lisbon), of 12 drivers in 24 different schedules, to Madrid.

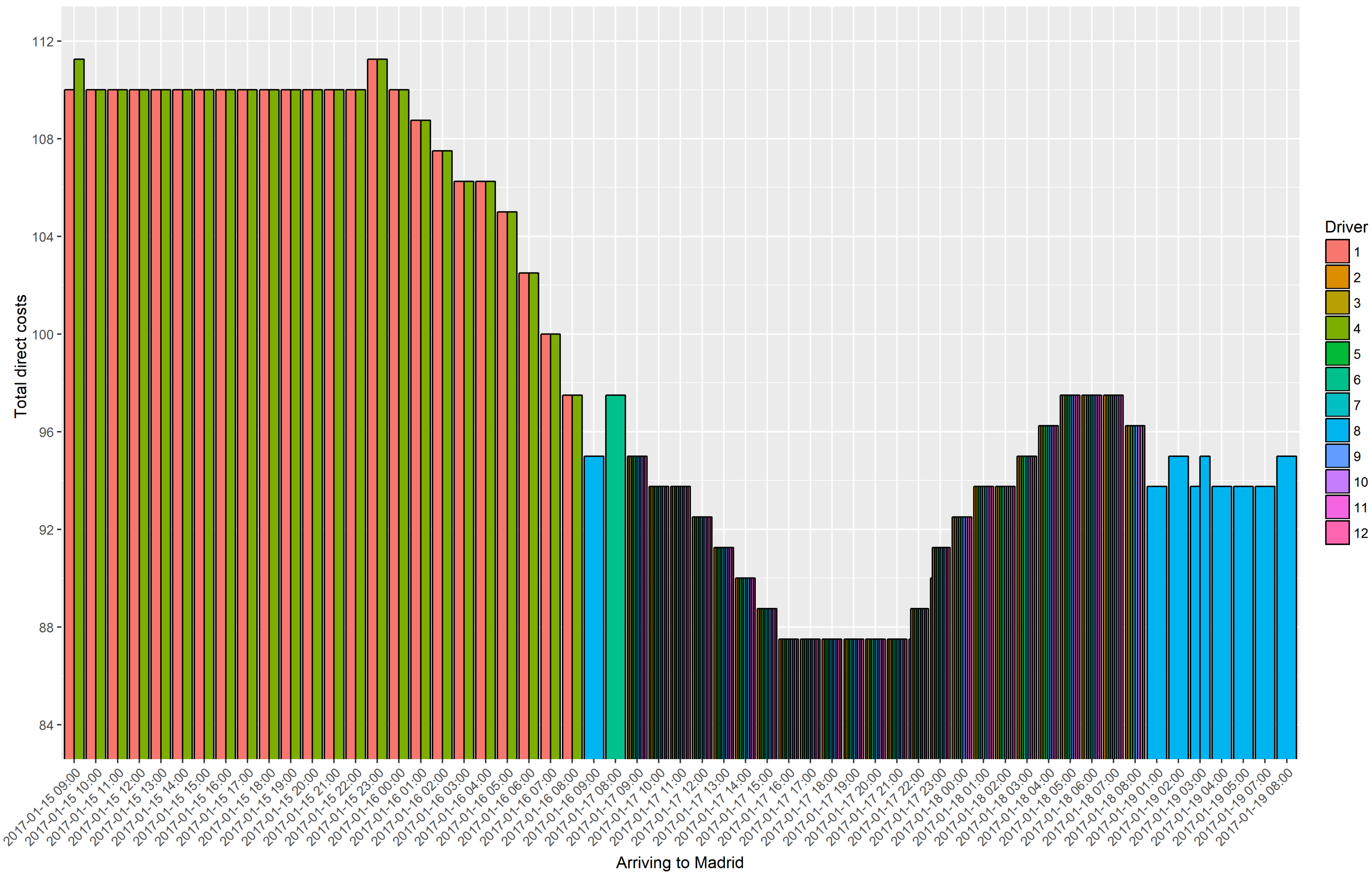

Figure 5.5: Plot of total direct costs by arriving (Madrid), of 12 drivers in 24 different schedules, to Madrid.

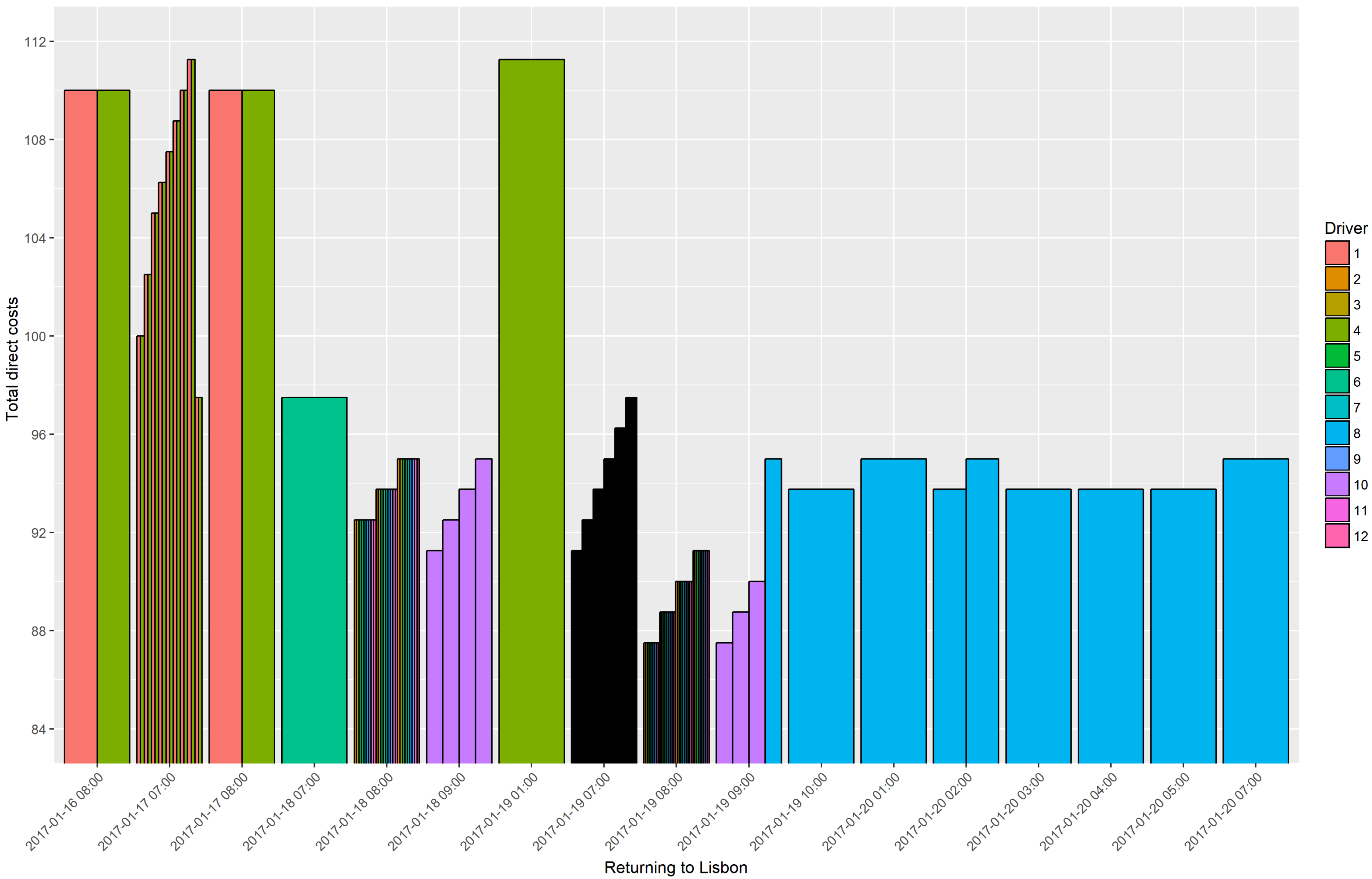

Figure 5.6: Plot of total direct costs by returning (back to Lisbon), of 12 drivers in 24 different schedules, to Madrid.

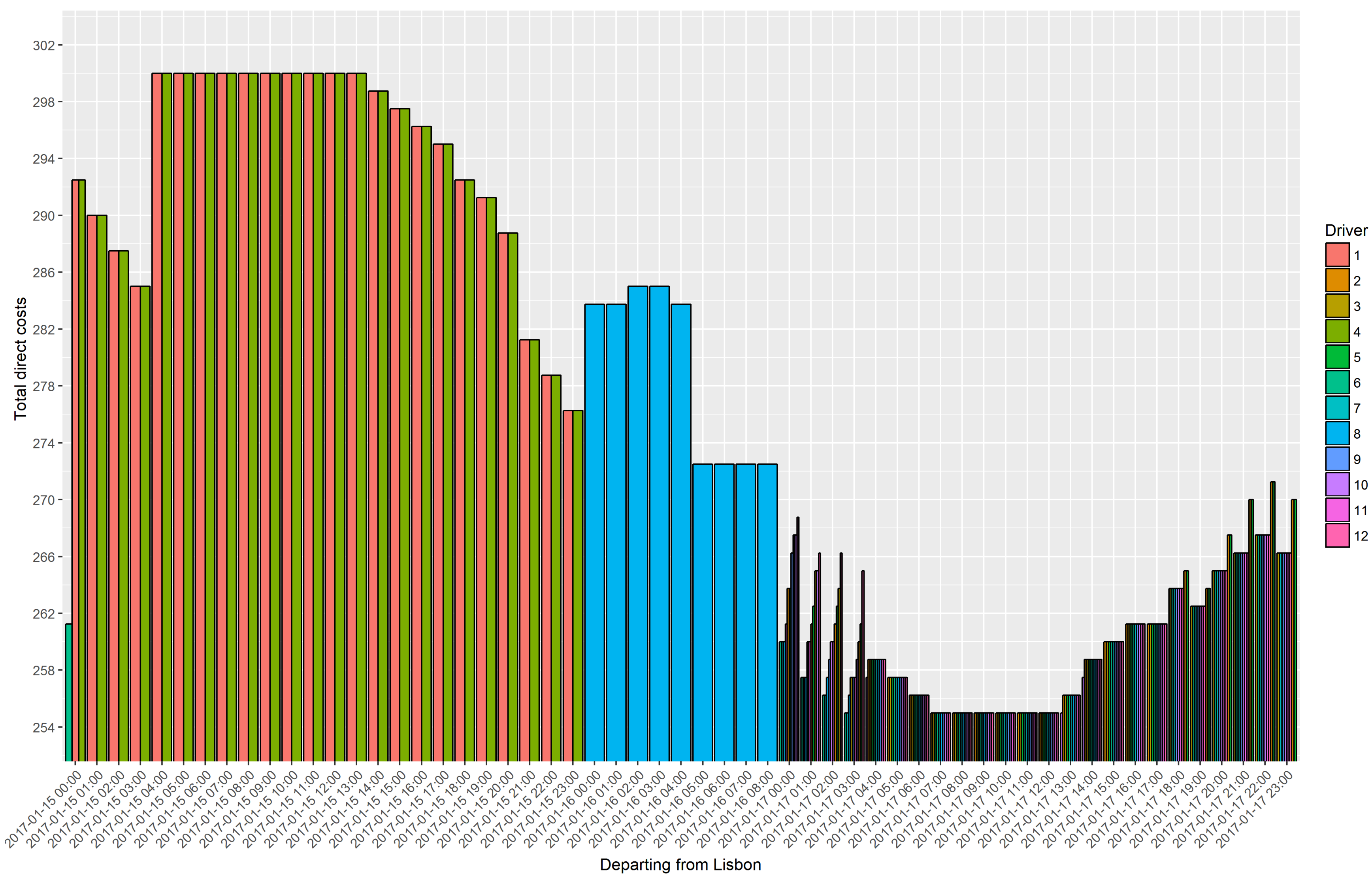

Figure 5.7: Plot of total direct costs by departing (Lisbon), of 12 drivers in 24 different schedules, to Paris.

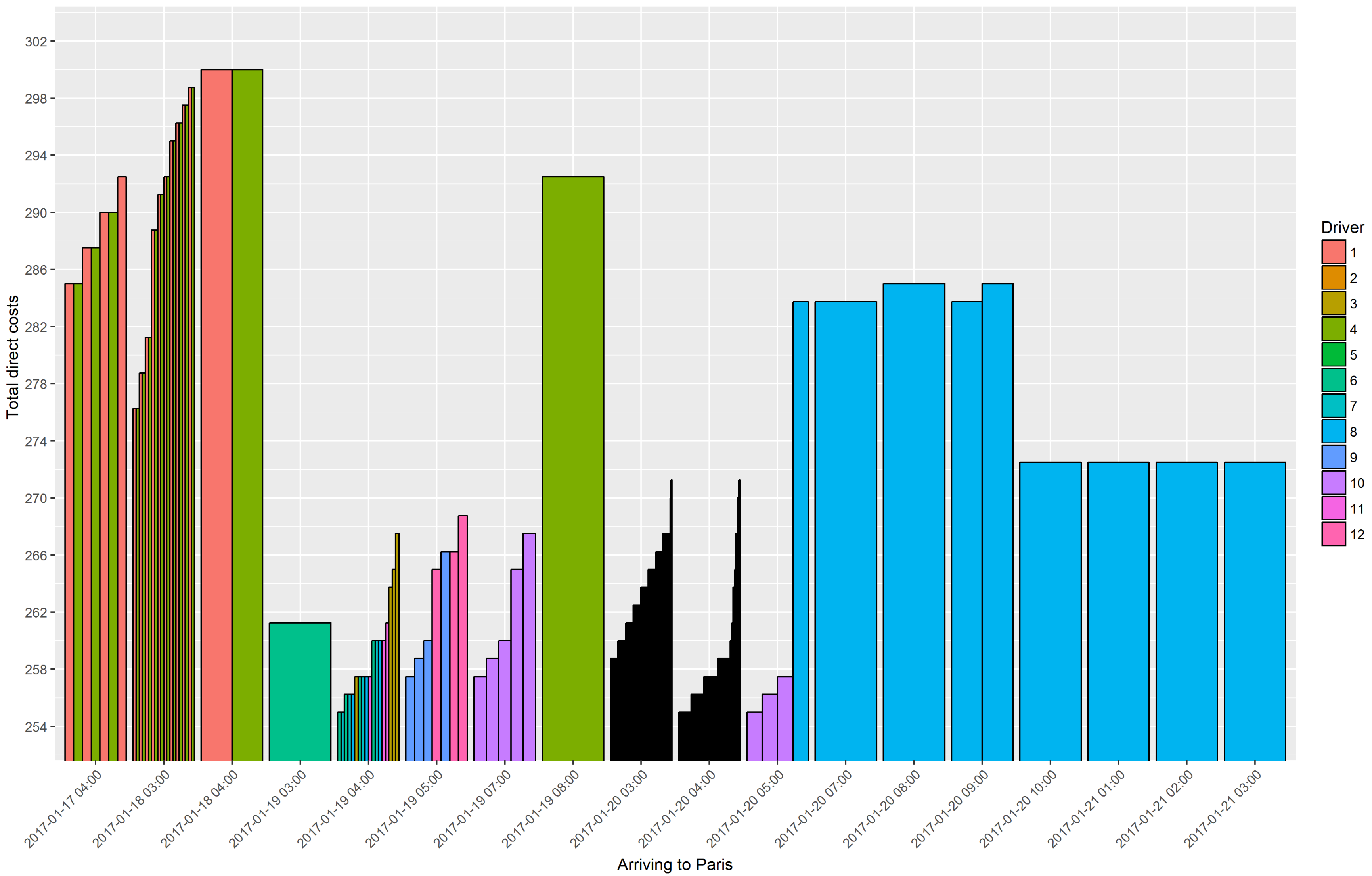

Figure 5.8: Plot of total direct costs by arriving (Paris), of 12 drivers in 24 different schedules, to Paris.

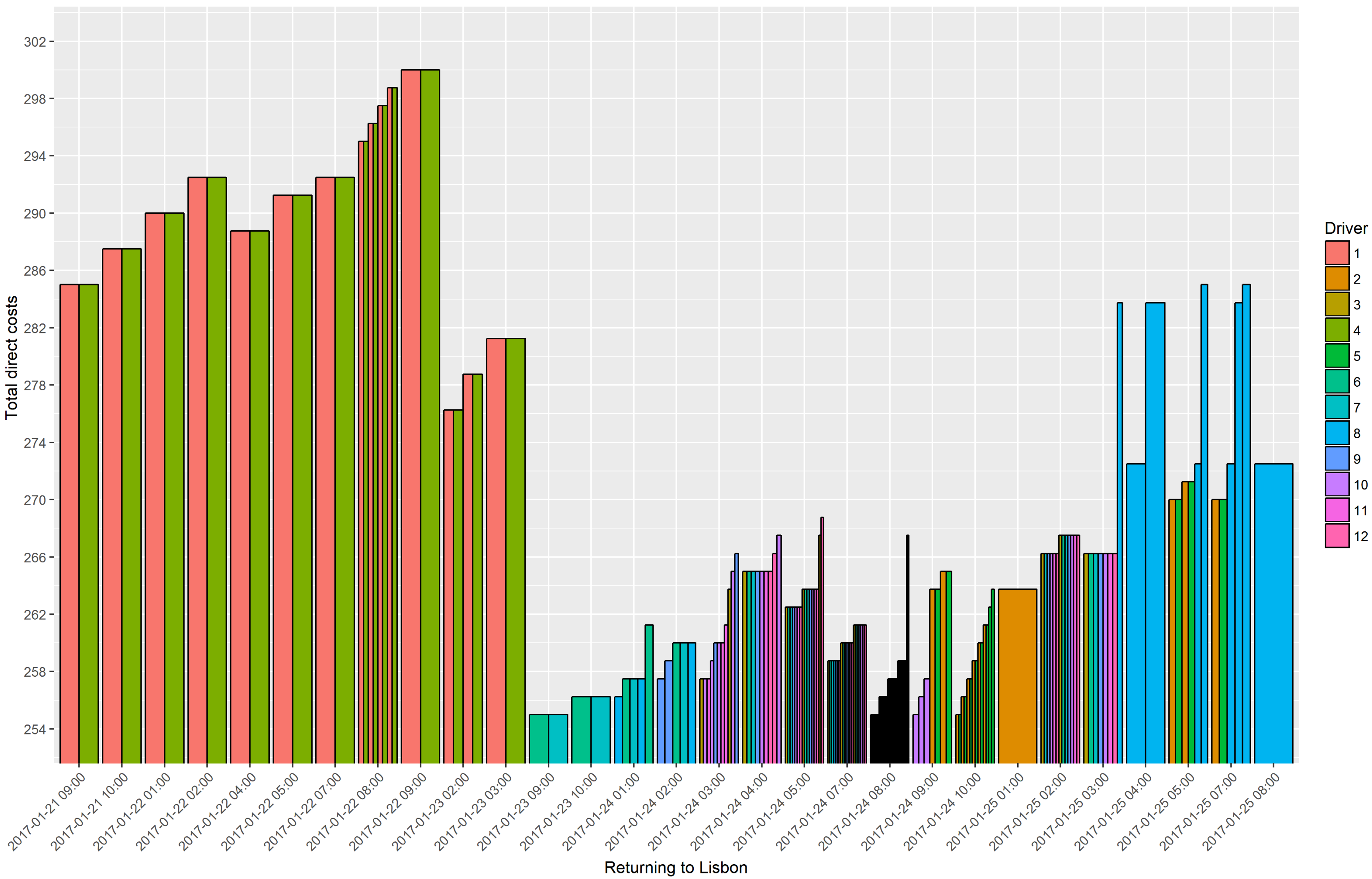

Figure 5.9: Plot of total direct costs by returning (back to Lisbon), of 12 drivers in 24 different schedules, to Paris.

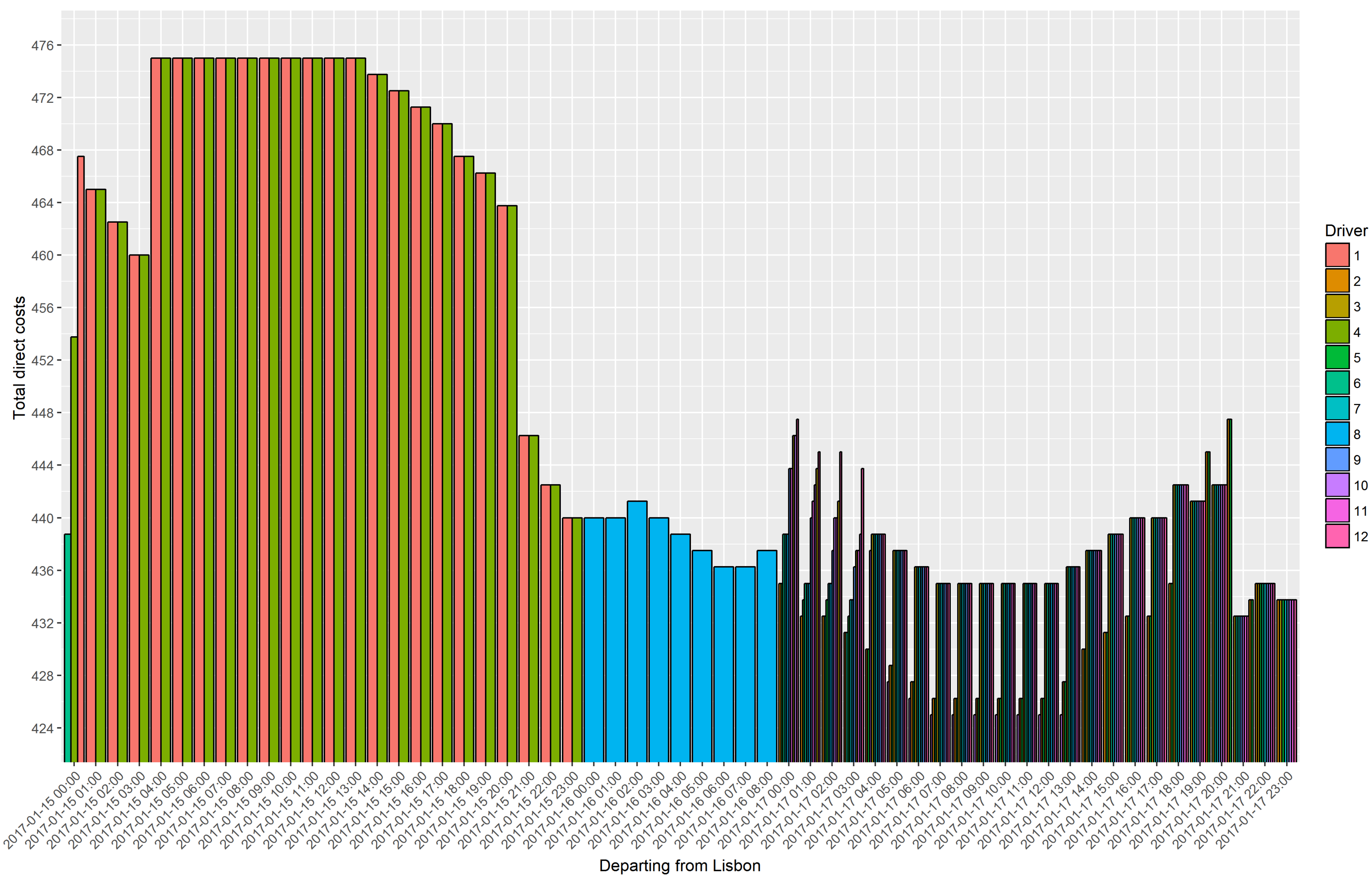

Figure 5.10: Plot of total direct costs by departing (Lisbon), of 12 drivers in 24 different schedules, to Berlin.

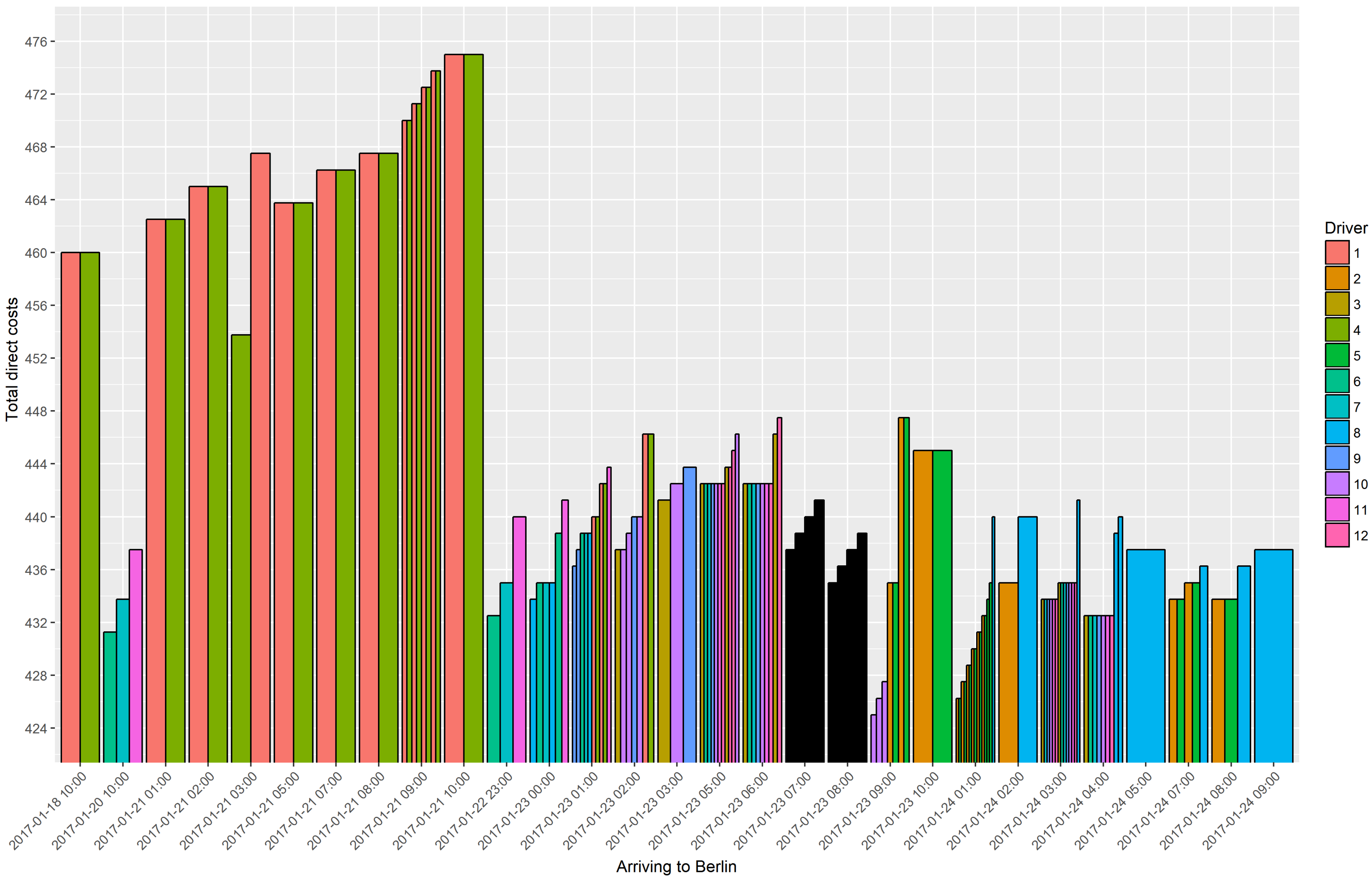

Figure 5.11: Plot of total direct costs by arriving (Berlin), of 12 drivers in 24 different schedules, to Berlin.

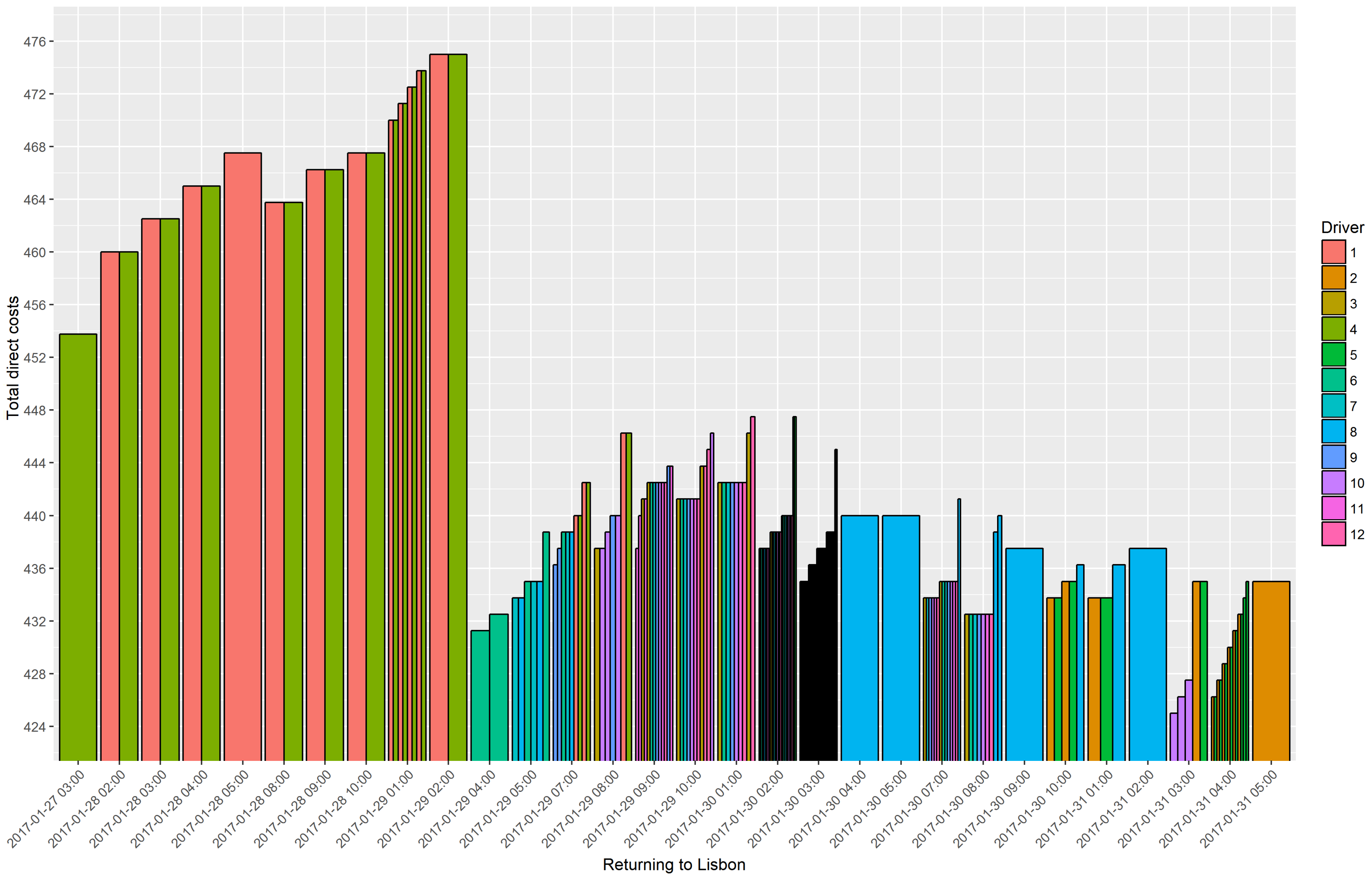

Figure 5.12: Plot of total direct costs by returning (back to Lisbon), of 12 drivers in 24 different schedules, to Berlin.

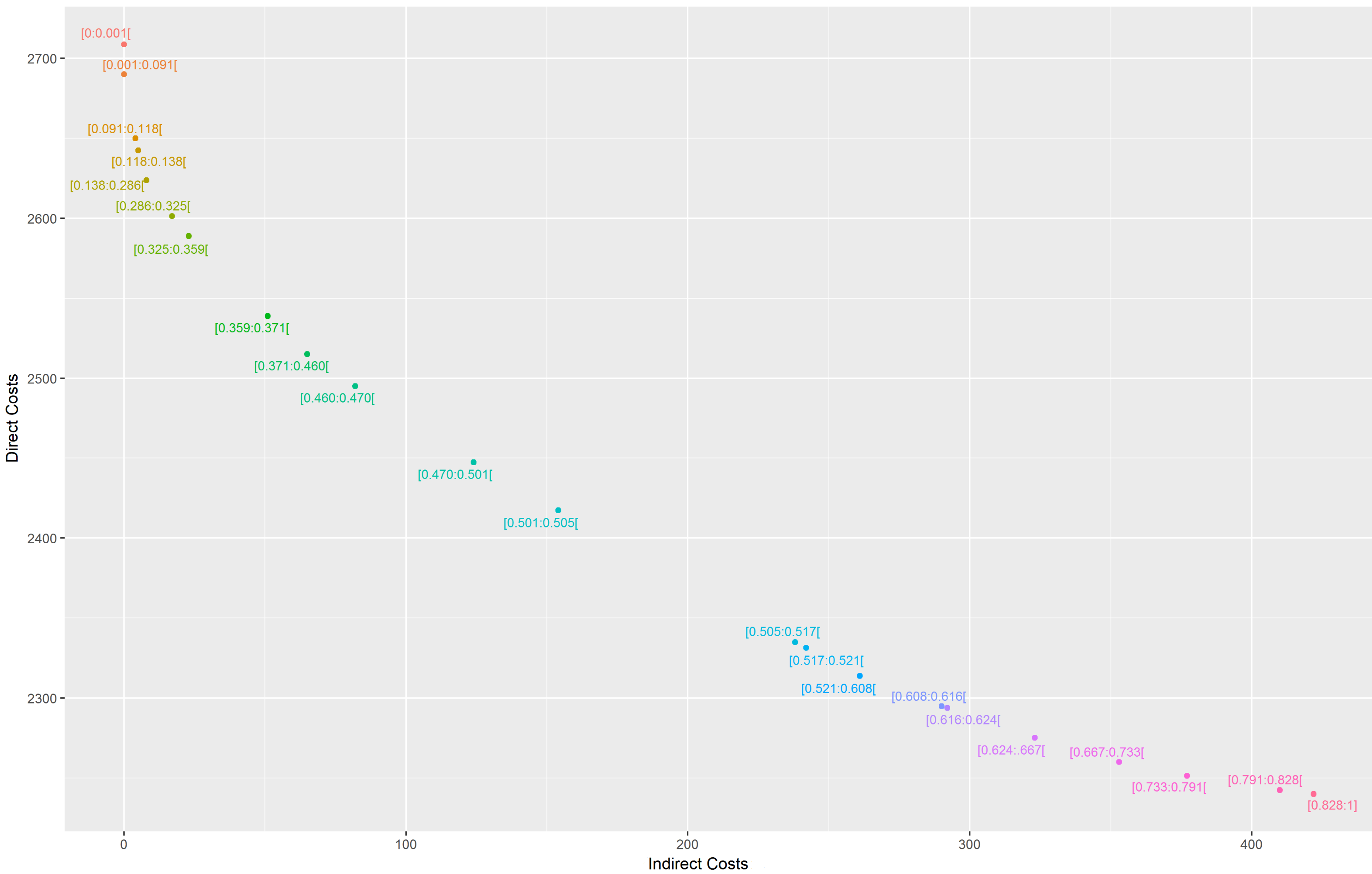

Figure 5.13: Plot showing how the impact on the optimization by changing the values of alpha.